



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**OPTIMIZATION OF AN ADVANCED MULTI-JUNCTION
SOLAR-CELL DESIGN FOR SPACE ENVIRONMENTS
(AM0) USING NEARLY ORTHOGONAL LATIN
HYPERCUBES**

by

Silvio Pueschel

June 2017

Thesis Advisor:

Sherif Michael

Co-Advisor:

Thomas Lucas

Second Reader:

Paul Sanchez

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.</p>			
1. AGENCY USE ONLY	2. REPORT DATE June 2017	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE OPTIMIZATION OF AN ADVANCED MULTI-JUNCTION SOLAR-CELL DESIGN FOR SPACE ENVIRONMENTS (AM0) USING NEARLY ORTHOGONAL LATIN HYPERCUBES		5. FUNDING NUMBERS	
6. AUTHOR(S) Silvio Pueschel			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number _____ N/A_____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>This thesis focuses on the replacement of a genetic algorithm currently used to optimize multi-junction solar cells with Silvaco Atlas simulation software. It introduces the nearly orthogonal Latin hypercube (NOLH) design of experiments (DoE) as a means for exploring and optimizing solar cell designs in Silvaco Atlas. The general applicability of this approach has been proven, and compared to the genetic algorithm optimization technique, the following achievements have been made. The pre-generated simulation designs can now be processed in parallel, which drastically reduces the time required to conduct multiple simulations. Moreover, the data generated using the NOLH enabled a better understanding of the simulation input/output relationship and helped to focus the solar cell development by highlighting the design parameters that matter most. Using stepwise regression to build a metamodel helped in finding an optimal design and revealing the interactions among the input parameters. The initial simulation has already yielded promising results and has clearly shown the preeminence of the NOLH over the genetic algorithm by identifying a design with greater than 21% more power output than in previous designs. The NOLH DoE should become the new standard for optimizing solar cells in Silvaco Atlas.</p>			
14. SUBJECT TERMS solar cell, InGaP, GaInP, GaAs, Ge, single cell, dual-junction cell, optimization, maximum power, efficiency, Silvaco ATLAS, simulation, photovoltaic, nearly orthogonal Latin hypercube, design of experiments			15. NUMBER OF PAGES 103
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**OPTIMIZATION OF AN ADVANCED MULTI-JUNCTION SOLAR-CELL
DESIGN FOR SPACE ENVIRONMENTS (AM0) USING NEARLY
ORTHOGONAL LATIN HYPERCUBES**

Silvio Pueschel
Lieutenant Colonel, German Army
Dipl.-Ing. (FH), University of German Armed Forces, Munich, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Sherif Michael
 Thesis Advisor

Thomas Lucas
Co-Advisor

Paul Sanchez
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis focuses on the replacement of a genetic algorithm currently used to optimize multi-junction solar cells with Silvaco Atlas simulation software. It introduces the nearly orthogonal Latin hypercube (NOLH) design of experiments (DoE) as a means for exploring and optimizing solar cell designs in Silvaco Atlas. The general applicability of this approach has been proven, and compared to the genetic algorithm optimization technique, the following achievements have been made. The pre-generated simulation designs can now be processed in parallel, which drastically reduces the time required to conduct multiple simulations. Moreover, the data generated using the NOLH enabled a better understanding of the simulation input/output relationship and helped to focus the solar cell development by highlighting the design parameters that matter most. Using stepwise regression to build a metamodel helped in finding an optimal design and revealing the interactions among the input parameters. The initial simulation has already yielded promising results and has clearly shown the preeminence of the NOLH over the genetic algorithm by identifying a design with greater than 21% more power output than in previous designs. The NOLH DoE should become the new standard for optimizing solar cells in Silvaco Atlas.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND.....	3
	A. GENERAL OVERVIEW SOLAR ENERGY USAGE.....	3
	1. Semiconductor Basics (Solar Cell).....	3
	2. Multi-junction Solar Cell.....	11
	3. Summary	13
	B. SILVACO ATLAS SIMULATION ENVIRONMENT	14
	1. Main Characteristics.....	15
	2. Emulating a Solar-Cell Structure	15
	3. Summary	17
	C. OPTIMIZATION.....	17
	1. Genetic Algorithm.....	17
	2. NOLH Design of Experiments	21
	3. Summary	26
III.	EXPERIMENTAL DESIGN	29
	A. EXPERIMENTAL SET UP.....	29
	1. Framework / Assumptions	30
	2. Design Creation	31
	3. Silvaco ATLAS Input File Setup.....	32
	B. EXPERIMENTAL EXECUTION.....	33
	1. Silvaco ATLAS Simulation Run	33
	2. Simulation Output.....	33
IV.	EXPERIMENTAL ANALYSIS	35
	A. EXPERIMENTAL ANALYSIS	35
	1. Agui Solar Cell with 257 Design Points.....	36
	2. Agui Solar Cell with 1,542 Design Points (Rotated and Stacked)	45
	3. Improved Agui Solar Cell with 1,542 Design Points (Rotated and Stacked)	50
	4. Second Iteration of Improved Agui Solar Cell with 1,542 Design Points (Rotated and Stacked).....	55
V.	CONCLUSION AND RECOMMENDATION.....	61

APPENDIX A. SILVACO ATLAS TEMPLATE FILE	65
APPENDIX B. PYTHON SCRIPTS.....	69
A. PYTHON SCRIPT FOR SILVACO ATLAS FILE PRE- PROCESSING	69
B. PYTHON SCRIPT TO RUN THE SILVACO ATLAS FILE.....	71
C. PYTHON SCRIPT FOR SILVACO ATLAS OUTPUT VALUE POST-PROCESSING.....	72
LIST OF REFERENCES	73
INITIAL DISTRIBUTION LIST	79

LIST OF FIGURES

Figure 1.	Atomic structure of intrinsic semiconductor materials Silicon and Germanium. Source: Physics and Radio-Electronics (2014a).....	4
Figure 2.	The Energy Band Diagram of a Semiconductor. Source: Hu (2009).....	5
Figure 3.	Phosphorus Doped Silicon Structure with Free Charge Carrier. Source: "Doping: N- and p-semiconductors" (n.d.).....	6
Figure 4.	Boron Doped Silicon Structure with Free New Hole. Source: "Doping: N- and p-semiconductors" (n.d.).....	7
Figure 5.	P-N Junction. Source: "Depletion region" (2015).	8
Figure 6.	Basic Structure of a Single Layer Solar Cell. Source: "Solar cell structure" (n.d.).....	9
Figure 7.	Solar Cell Efficiency by Material. Source: "Solar cell efficiency limit" (n.d.).	10
Figure 8.	Current vs. Voltage (Green) and Power vs. Voltage (Blue) for a Solar Cell at a Number of Different Illumination Levels. Source: Celani (2013).....	11
Figure 9.	Solar Spectrum vs. Material and General Buildup of Multi-Junction Solar Cell. Source: Eurems (n.d.).....	11
Figure 10.	Efficiency Chart by Solar Cell Design. Source: National Renewable Energy Laboratory (2017).	13
Figure 11.	Example 32-bit Chromosome Genetic Representation of a Semiconductor Application. Source: Bates (2004).....	18
Figure 12.	Example of 1% Mutation of Genes in Chromosome. Source: Bates (2004).....	19
Figure 13.	Example Dual-Point Crossover of Genes in Chromosome. Source: Bates (2004).....	19
Figure 14.	Research Steps Toward the NOLH DoE. Source: MacCalman (2013).....	23
Figure 15.	Mathematical Metamodel of NOLH DoE Simulation Result Analysis. Source: MacCalman (2013).....	24

Figure 16.	Experimentation Setup Schematic.....	30
Figure 17.	Current vs. Voltage (I-V Curves) and Output Power vs. Voltage Generated from the Maximum Output Power Simulation Run.....	34
Figure 18.	Extract of Scatterplot Matrix of 257-Design-Point Agui Cell Design Using Logarithmic-Transformed Dopants.	37
Figure 19.	Extract of Scatterplot Matrix of 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	38
Figure 20.	Simulation Output Power Distribution of the 257-Design-Point Agui Cell Using Linear-Transformed Dopants.....	40
Figure 21.	Prediction Profiler for Response Surface of 257-Design-Point Agui Cell Design Optimization.	43
Figure 22.	Extract of Scatterplot Matrix of 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	46
Figure 23.	Simulation Output Power Distribution Improved in 1,542-Design-Point Agui Cell Design Using Linear Transformed Dopants.....	46
Figure 24.	Prediction Profiler for Response Surface Improved 1,542-Design-Point Agui Cell Design Optimization for Linear-Transformed Dopants.....	50
Figure 25.	Extract of Scatterplot Matrix for Improved 1,542-Design-Point Agui Cell Design Using Linear Transformed Dopants.....	51
Figure 26.	Simulation Output Power Distribution for Improved 1,542-Design-Point Agui Cell Design Using Linear Transformed Dopants.....	52
Figure 27.	Residual by Prediction Plot for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	54
Figure 28.	Prediction Profiler for Response Surface Improved 1,542-Design-Point Agui Cell Design Optimization Using Linear-Transformed Dopants.....	55
Figure 29.	Extract of Scatterplot Matrix from Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	56

Figure 30.	Simulation Output Power Distribution for Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	57
Figure 31.	Prediction Profiler for Response Surface in Second Iteration of Improved 1,542-Design-Point Agui Cell Design Optimization Using Linear-Transformed Dopants.....	59
Figure 32.	Current vs. Voltage (I-V Curves) for the NOLH Basic and Optimized Two-Junction Solar-Cell Designs Generated from the Maximum Output Power Simulation Run.	60

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Head Snapshot of NOLH Design Generating Matrix.....	25
Table 2.	NOLH Excel Header with Some Example Values.....	31
Table 3.	Correlation Matrix of Input Parameters and Output Power Values of 257-Design-Point Agui Cell Design Using Linear- Transformed Dopants.....	39
Table 4.	Effect Summary Model Fit for 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	41
Table 5.	Summary of Fit Model of 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	42
Table 6.	Effect Summary Model Fit for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	48
Table 7.	Summary of Fit Model for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	49
Table 8.	Effect Summary Model Fit Improvements for 1,542-Design- Point Agui Cell Design Using Linear-Transformed Dopants.....	53
Table 9.	Summary of Fit Model Improvements for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	54
Table 10.	Effect Summary Model Fit for Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear- Transformed Dopants.....	58
Table 11.	Summary of Fit Model for Second Iteration of Improved 1,542- Design-Point Agui Cell Design Using Linear-Transformed Dopants.....	59

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CCD	central Composite Design
DoE	Design of Experiments
GA	genetic algorithm
I-V curve	Current vs. Voltage curve
LHS	Latin hypercube sampling
NOB	nearly orthogonal balanced
NOLH	nearly orthogonal Latin hypercube
NPS	Naval Postgraduate School
OLH	orthogonal Latin hypercube
SEED	Simulation, Experiments, and efficient Design

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This thesis applies the nearly orthogonal Latin hypercube (NOLH) design of experiments (DoE) to optimize the design of multi-junction solar cells with Silvaco Atlas simulation software. The results show that the output power of a dual-junction solar cell can be increased by 21%.

In previous research, a genetic algorithm was used to optimize the design of multi-junction solar cells in conjunction with Silvaco Atlas. The process suffered from the following limitations:

- It was complex to set up and inflexible to cell design changes as well as produced an insufficient resolution;
- Because only sequential simulation runs were possible, long run times were required; and
- The process produced an estimated optimum, but no data were available in a form that facilitated understanding of the relationships between inputs and output.

Given the aforementioned restrictions, the potential for that approach in the development of more complex multi-junction solar cells seemed exhausted.

The introduction of the NOLH addresses all limitations and achieves the following improvements to the development process:

- Design preparation is easy and flexible using the NOLH Excel spreadsheet;
- Higher resolution is easy through rotation and stacking;
- Pre-generating all input combinations allows parallel processing of the Silvaco Atlas simulation files (future implementation); and
- Data analysis identifies relevant input parameters and their behaviors to focus cell design development.

With the NOLH properties described, the limitations of the genetic algorithm can be mitigated and the development turnaround times reduced by an order of magnitude. The new process explores the experimentation space with a higher resolution and detects maximum output values with greater accuracy.

With the new process, solar cell developers have a powerful, efficient tool to focus their research on the solar cell design, no longer limited by the optimization process alone. The following table gives an overview of the input parameters used to obtain the highest power output.

Design parameter	Basis Design	Lin basis	Lin stacked	Second iteration	Third iteration
winthick	0.33	0.011	0.012	0.007	0.00134
windopconc	2.00e+18	1.0e+19	8.72e+18	5.68e+18	6.13E+18
emthick	0.05	0.024	0.067	0.025	0.0328
emdopconc	2.00e+18	2.42e+18	6.79e+18	3.2e+17	1.50E+17
basethick	0.55	0.343	0.338	0.284	0.357
basedopconc	1.50e+17	3.3e+17	2.4e+17	8.72e+18	4.30E+18
bsfthick	0.03	0.015	0.039	0.034	0.026
bsfdopconc	2.00e+18	6.4e+18	1.65e+18	2.9e+17	1.10E+17
buffthick	0.03	0.04688	0.01391	0.044	0.02746
buffdopconc	1.00e+18	6.75e+18	6.94e+18	8.28e+18	9.44E+18
tunemthick	0.015	0.019	0.013	0.011	0.0131
tunemdopconc	8.00e+18	6.76e+19	3.87e+19	3.8e+17	7.85E+18
tunbasesthick	0.015	0.013	0.016	0.02	0.03
tunbasedopconc	1.00e+19	8.35e+18	7.41e+19	5.29e+18	8.00E+18
botwinthick	0.05	0.036	0.021	0.016	0.0105
botwindopconc	1.00e+19	1.77e+18	9.46e+19	1.21e+18	2.44E+18
botemthick	0.1	0.079	0.06	0.126	0.1222
botemdopconc	2.00e+18	1.8e+18	1.88e+18	5.7e+17	1.98E+17
botbasesthick	3	4.8	4.8	4.2	5.27
botbasedopconc	1.00e+17	4.2e+17	2.1e+17	4.22e+16	1.88E+16
botbsfthick	0.1	0.106	0.097	0.127	0.1159
botbsfdopconc	2.00e+18	3.7e+17	9.38e+18	1.4e+17	4.72E+17
botbuffthick	0.3	0.422	0.325	0.368	0.3422
Botbuffdopconc	7.00e+18	3.61e+18	1.18e+19	1.09e+19	4.01E+19
Max power output	35.76	39.58	40.38	42.07	42.86

The starting point of the optimization process, called the basis design, was a dual-junction design optimized by the genetic algorithm for an AM1.5 spectrum. The most significant improvement from the basis design to the linear basis is explained by the first use of the AM0 spectrum. The linear basis is the first simulation run of the Silvaco Atlas for a design with NOLH-generated input variables. As shown in the table, the Lin stacked and the second iteration output values showed significant improvement over the Lin basis output value. This improvement resulted from better coverage of the experimentation space for the Lin stacked design and adjusted input parameter ranges for the second iteration

design. The adjustment procedure was refined in the third iteration and demonstrated a successful application of the method to improve cell design in Silvaco Atlas.

The results of the simulations highlight the potential of the NOLH DoE. It is recommended that the NOLH replace the genetic algorithm as the new standard for developing solar cells with Silvaco Atlas.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Professor Sherif Michael for his trust and the opportunity to test this very new direction of research. I would also like to thank Professor Thomas Lucas and Professor Paul Sanchez for their advice and guidance in analyzing data and for keeping me on track in my research. Special thanks goes to Matthew Porter; he was a great help in all issues concerning the Silvaco Atlas simulation environment and other related questions.

Finally, I sincerely thank my wife, Christin, and my son, Caspar, for their unwavering support, inexhaustible patience, and understanding with the many hours of effort required to complete this research.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The use of electrical solar energy dates back to 1839, when Alexandre Edmond Becquerel first observed the photovoltaic effect. This discovery marked the start of a long journey to improve the efficiency and make the effect and its application a commercial success. An important step along the way was the 1883 construction of the first solid-state solar cell by Charles Fritts using selenium. In 1887, Heinrich Hertz discovered the photoelectric effect, which Albert Einstein theoretically explained in 1905 on a quantum basis. From 1910 to 1940, several achievements in the field of semiconductor material were made, leading to the development of the first silicon-based solar cell in 1941 by Russell Ohl. In the following years, efficiency improved, as demonstrated in the first U.S. solar-powered satellite, Vanguard I, in 1958 (U.S. Department of Energy, n.d.).

Today, we have to distinguish between applications for commercial use, which are mostly silicon-based, and applications with higher efficiency obtained by using combinations of more exotic semiconductor materials such as gallium arsenide (GaAs) or gallium indium phosphate (GaInP). Due to the higher costs, solar cells consisting of these latter materials are more likely used in special applications, such as space operations, for which launch costs are a primary concern. The goal is to reduce the size of the solar panel and, subsequently, the weight, while generating the same amount of electrical power.

This is achieved by researching substances and their photovoltaic behavior. Another promising approach is to experiment with multi-junction solar cells to improve the yield per photosensitive area by stacking different materials on each other. Multi-junction cells are able to use a wider spectrum of light than single-junction cells to produce electrical power. To develop such cells, a design method to build and test the cells is used. Due to the complexity of the buildup, this process is costly and time-consuming. A better, more cost-efficient way is to simulate the electrical behavior of such designs and improve their performance

by applying optimization algorithms. Such a computer model, able to accurately model multi-junction solar cells and predict their performance, is discussed in this thesis. The emphasis of the work lies in the introduction of a new technique to optimize the cell design rather than the optimization of the cell itself.

This thesis builds on a long series of research on this subject, which started with Michalopolous (2002) and Green (2002), to model solar cells with the Silvaco ATLAS simulation environment. Theses by Bates (2004a) and Utzler (2006) improved the cell design with a genetic algorithm and a compendium of the prior work, while the dissertation by Tsutagawa (2013) refined the processes.

Building on this earlier work, this thesis replaces the genetic algorithm with the nearly orthogonal Latin hypercube (NOLH) design of experiments (DoE) developed by Cioppa and Lucas (2007), Hernandez, Lucas, and Carlyle (2012), and MacCalman, Vieira, and Lucas (2017). The goal is to show the usefulness of this approach and to analyze whether the results produced are as good as or even better than the ones generated by the genetic algorithm. It is worth mentioning that the genetic algorithm and NOLH yield improved solutions to an optimization problem rather than optimize a solution in a mathematical sense.

The outline of this thesis is as follows. Chapter II provides the background of semiconductor physics and its application to solar cells, introduces the main characteristics of the Silvaco ATLAS simulation environment, and describes the previous optimization algorithm and the novel approach using NOLHs as well as the advantages and disadvantages of the latter. Chapter III introduces the experimental setup and highlights the results using the statistical analysis tool JMP (SAS Institute, n.d.-c). Chapter IV discusses and compares the results with previous research, specifically the dual-junction cell first introduced by Japan Energy Corporation's Central Research Laboratory (Agui, Takamoto, Ikeda, and Kurita, 1998). This study was used as a reference by Tsutagawa (2013) to validate the Silvaco ATLAS simulation capability for use with the genetic optimization algorithm. Chapter V draws conclusions and makes recommendations for future research.

II. BACKGROUND

This thesis is follow-on research to a sequence of efforts to optimize the energy output of multi-junction solar cells with the Silvaco ATLAS simulation environment (SILVACO Inc., n.d.). For a better understanding of the subject and to provide the basis for solar cell physics, this chapter first introduces semiconductor materials and their application in solar cells. Next, the previous genetic algorithm optimization technique and the replacement NOLH approach are discussed. The goal of this chapter is to highlight the benefit of the new approach in the design development of multi-junction solar cells with Silvaco ATLAS.

A. GENERAL OVERVIEW SOLAR ENERGY USAGE

Energy from the sun has become a vital part in the power-supply strategy for many countries and is the preferred source of energy for space operations. This thesis focuses on solar cells used in space applications because multi-junction solar cells are currently too expensive for general commercial use.

1. Semiconductor Basics (Solar Cell)

In the world of electricity, there are two main types of materials, ones that conduct electricity freely, called conductors, and ones that have no freely moving electrons, called non-conductors or insulators. Semiconductor materials, as the name suggests, are somewhere in the middle; they are neither a true conductor nor an insulator. The most important materials that belong to this group include germanium, silicon, gallium arsenide, and a variety of other substances or their combinations. All these substances are crystalline inorganic solids. A different area of research is ongoing to investigate the properties of organic materials and their use in solar cell applications.

The sole focus of this thesis is on inorganic materials. Each of these materials has very specific properties and, therefore, the understanding of its

behavior under certain environmental conditions is vital for the development of specific electronic applications such as solar cells. The most important physical properties are covered in the following paragraphs to explain the specific substance behavior.

a. **General physics**

Every chemical substance or semiconductor material structure can be described using the Bohr atomic model. The model consists of an atom comprised of a small, positively charged nucleus surrounded by free movable electrons that travel in orbits, or shells, around the center. The distance of the circular orbit from the center is defined by the electron's energy levels. As depicted in Figure 1, for example, silicon has 14 electrons total in its neutral state with four valence electrons in its outer shell. This classifies silicon as a Group IV element in the Periodic Table of Elements and as an intrinsic semiconductor (Periodic Table of Elements, 2014).

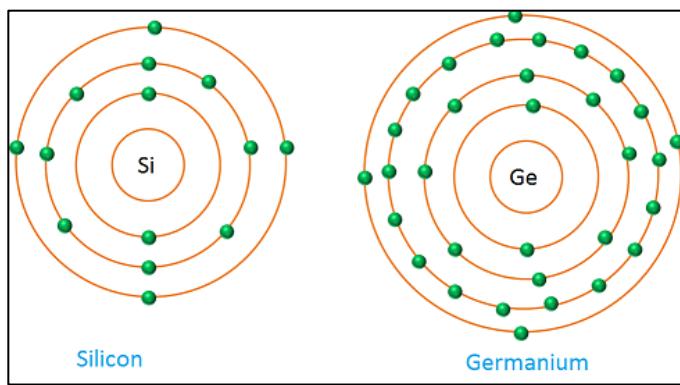


Figure 1. Atomic structure of intrinsic semiconductor materials Silicon and Germanium. Source: Physics and Radio-Electronics (2014a).

For a better understanding of the electrical properties of a semiconductor, the following paragraphs provide a brief discussion of the most important characteristics related to the technical application of solar cells.

b. Material Bands and their Energy levels

The concept of energy bands is most useful for describing the electrical mechanism of semiconductor materials. According to the Pauli exclusion principle, “No two electrons in an atom can be at the same time in the same state or configuration” in an electron system such as a semiconductor crystal (“Pauli Exclusion Principle”, 2016). In such a crystalline material, many atoms are brought into proximity, which results in replacing the discrete energy levels with bands of energy states. Those states are separated by a gap as illustrated in Figure 2.

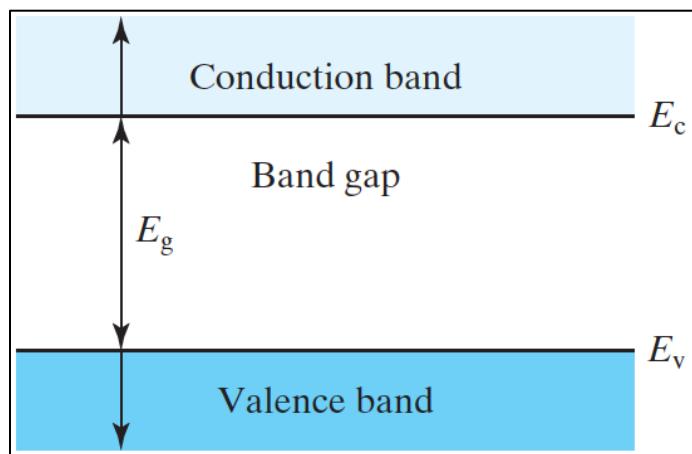


Figure 2. The Energy Band Diagram of a Semiconductor.
Source: Hu (2009).

Naturally, materials tend to equalize the different energy levels. However, in semiconductors, the lower energy bands are filled completely (as depicted by the valence band, energy level E_v) whereas the higher energy bands are completely empty (as depicted by the conduction band, energy level E_c). The region in between is called the band gap and does not allow conduction of current. The difference of the energy levels E_c and E_v is the energy gap E_g . Clearly, $E_g = E_c - E_v$, which is 1.1 eV for silicon. The number of free electrons (the valence band) and the holes (the conduction band) can change based on

the presence of impurities in the material called dopants. (Physics and Radio-Electronics, 2014b)

c. Doping

Doping is the process of adding a small amount of impurity to the intrinsic semiconductor materials, making the impurity extrinsic. This process modifies the electrical characteristics of the basic intrinsic semiconductor material. The dopant is integrated into the atomic structure of the semiconductor crystal, so the number of outer electrons defines the type of doping. Silicon's most used dopants are boron (three valence electrons) and phosphorus (five valence electrons). The element with three valence electrons is used for p-type doping; the substance with five valence electrons is used for n-type doping. Physics and Radio-Electronics, (2014b).

N-type Doping: By introducing a five-valent dopant, four of the outer silicon electrons combine with one of the dopant electrons while the fifth electron moves freely and serves as a charge carrier (see Figure 3). Compared to intrinsic silicon, doped crystal materials need much less energy to overcome the band gap and move the free electrons from their valences into the conduction bands. The doping substance, which is used for n-doping, is known as an electron donor.

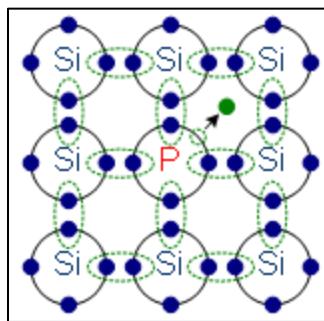


Figure 3. Phosphorus Doped Silicon Structure with Free Charge Carrier.
Source: "Doping: N- and p-semiconductors" (n.d.).

P-type Doping: Three valence electrons have the opposite effect of five-valent doping. Having only three valence electrons to combine with the outer silicon electrons leaves a hole instead of a free electron in the valence band of the silicon (see Figure 4). N-doping in one material and P-doping in another ensures the reduction of the energy level necessary to move an electron from the valence band into the conduction band. Hence, the material's resistance lowers, so current is able to flow. Due to the controlled contamination of silicon material, the conductivity of the material can be increased by a large factor. (Doping: N- and p-semiconductors, n.d.).

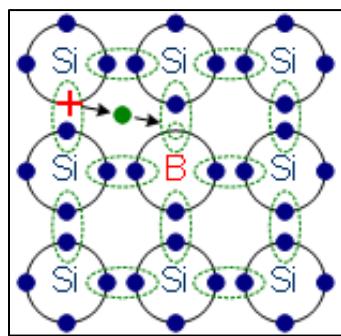


Figure 4. Boron Doped Silicon Structure with Free New Hole.
Source: "Doping: N- and p-semiconductors" (n.d.).

d. **P–N Junction**

The area where the n- and p-doped semiconductor materials come together is called the p-n junction. At the thermal equilibrium, there are no free charge carriers in this transition area. The free electrons from the n-doped region have recombined with the holes in the p-doped part of the material (see Figure 5) with the effect that no current flows if no external energy is applied to the semiconductor materials.

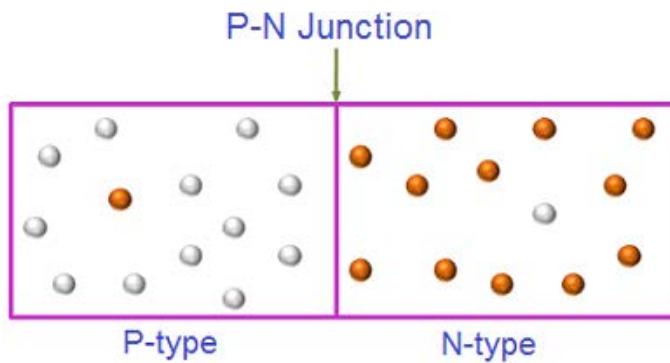


Figure 5. P-N Junction. Source: “Depletion region” (2015).

Due to the loss of free charge carriers—a positive charge in the n-doped region or a negative charge in the p-doped region—an area of differing energy level exists. Hence, a depletion zone forms at the interface between the two doped materials. The width of the depletion zone depends on the recombination of holes and electrons and is defined by the basic semiconductor material and the doping substances. The recombination comes to a total standstill because the free electrons and holes can no longer overcome the electric field.

e. *Solar Cell Operation*

As depicted in Figure 5, the p-n junction forms the basis of a single layer solar, or photovoltaic, cell. The cell is an electrical device that is able to transform the energy of light directly into electricity using the photovoltaic effect.

The solar cell generally comprises three active layers. The top junction layer made of n-type semiconductor material is called the emitter layer, which emits solar energy—freed electrons into the p-n junction. The p-n junction region, called the absorber layer, is the depletion zone that provides the electrical voltage due to the electric field created by the different energy levels (as described in the previous paragraph d) and forces the freed electrons into a current. The back layer is made of p-type material, which provides the holes to absorb the electrons emitted from the n-type material. The front metal grid and the back metal plate ensure the flow of the collected current. To get the current to flow, front and rear

electrodes have to be attached to the surface of the semiconductor material, as shown in Figure 6 (“Solar cell structure” n.d.).

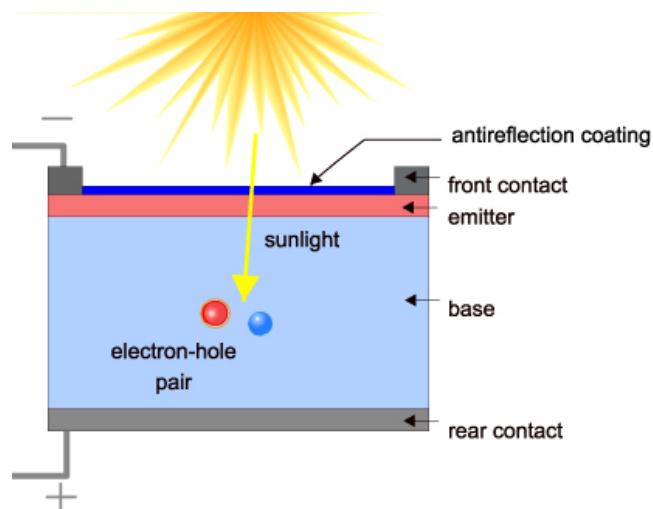


Figure 6. Basic Structure of a Single Layer Solar Cell.
Source: “Solar cell structure” (n.d.).

f. **Solar Cell Parameter**

The efficiency of solar cells depends heavily on the material and its combination, the doping material, and the spectrum of light to which the cell is exposed. Other parameters affecting the efficiency include reflectance, thermodynamics, charge carrier separation, and conductive efficiency according to Bates (2004b). In this thesis, they are not considered.

According to Shockley and Queisser (1961), there exists a maximum theoretical efficiency limit for a single solar cell derivable from the band gap and the spectrum of light (see Figure 7). They conclude that semiconductors with a band gap between 1 and 1.5 eV, which corresponds with a nearly infrared light spectrum, have the highest efficiency potential among single-junction solar cells.

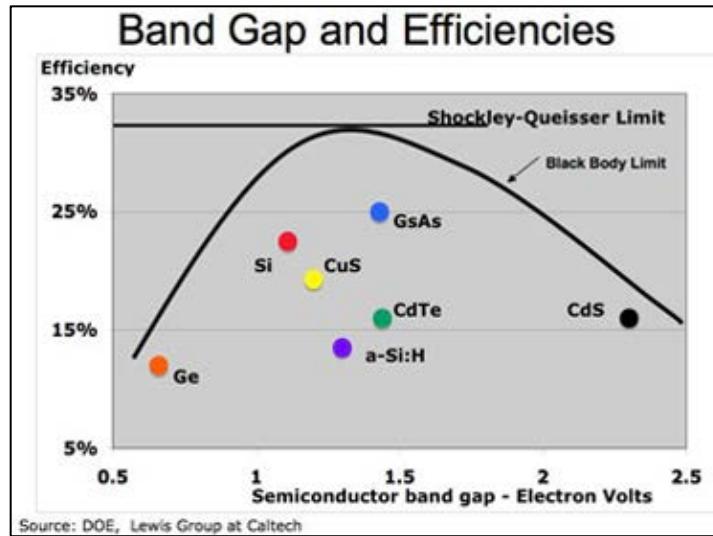


Figure 7. Solar Cell Efficiency by Material. Source: “Solar cell efficiency limit” (n.d.).

Since the light spectrum is different under specific environmental and geographical conditions, solar cells have to be optimized for these settings. Hence, the material used for the solar cell and its fabrication becomes a vital part of optimizing efficiency. Optimizing a solar cell entails keeping illumination at a steady level as a fixed parameter and varying the material type, thickness, and doping levels. The P_{Panel} (W) graph in Figure 8 shows the power output as the gauge of efficiency for definite levels of illumination.

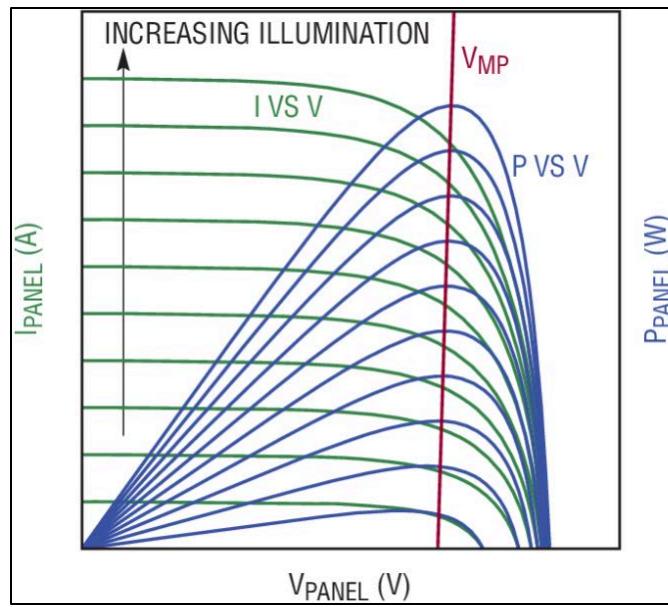


Figure 8. Current vs. Voltage (Green) and Power vs. Voltage (Blue) for a Solar Cell at a Number of Different Illumination Levels.
Source: Celani (2013).

2. Multi-junction Solar Cell

Recent developments in high-efficiency multi-junction solar cells take advantage of different semiconductor substances using only certain ranges of the light spectrum to generate electrical power (see Figure 9, right illustration).

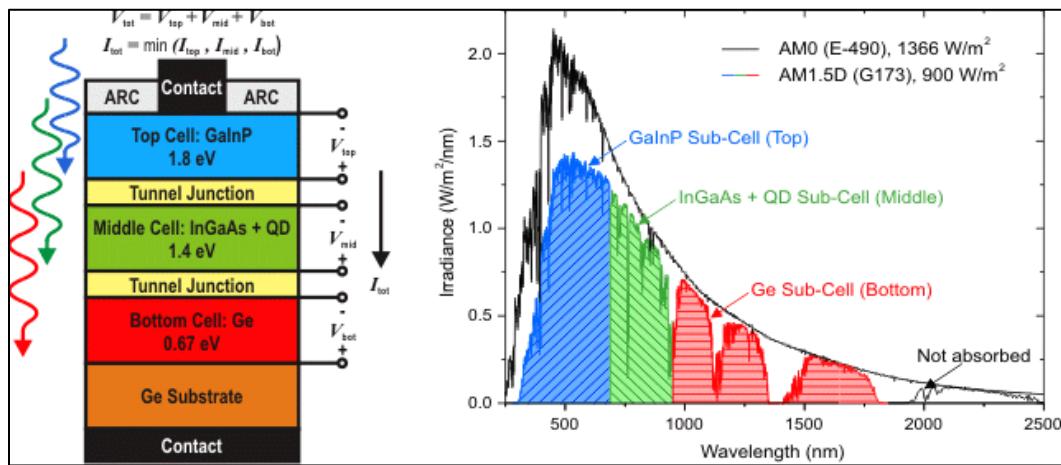


Figure 9. Solar Spectrum vs. Material and General Buildup of Multi-Junction Solar Cell. Source: Eurems (n.d.).

Stacking single-junction cells in series builds a bigger area to generate energy. Each layer has a different band gap. Therefore, each layer absorbs photons from light that has greater energy than the band-gap layer (see Figure 9, left illustration). Photon energy E and wavelength λ are inversely related by

$$E = \frac{hc}{\lambda} , \quad (1)$$

$$E[\text{eV}] \approx \frac{1.24}{\lambda[\mu\text{m}]} , \quad (2)$$

where h is Planck's constant ($4.136 \times 10^{-15} \text{ eV}\cdot\text{sec}$) and c is the speed of light. As shown in Figure 9, a highly doped, thin tunnel junction is placed between consecutive p-n junction layers to reduce the forward bias voltage drop from the normal p-n junction. Another limiting factor in multi-junction solar cells is the flow of the current. To avoid a current bottleneck, constructive measures are applied to balance the resistance of the overall structure and the solar physics to yield as much electricity from the sun as possible. The theoretical efficiency of stacking an infinite number of single-junction cells has a limit of 86.8% (Green, 2003, p. 65). Current research has achieved a maximum of 46% efficiency for a concentrated four-junction cell, as shown in Figure 10.

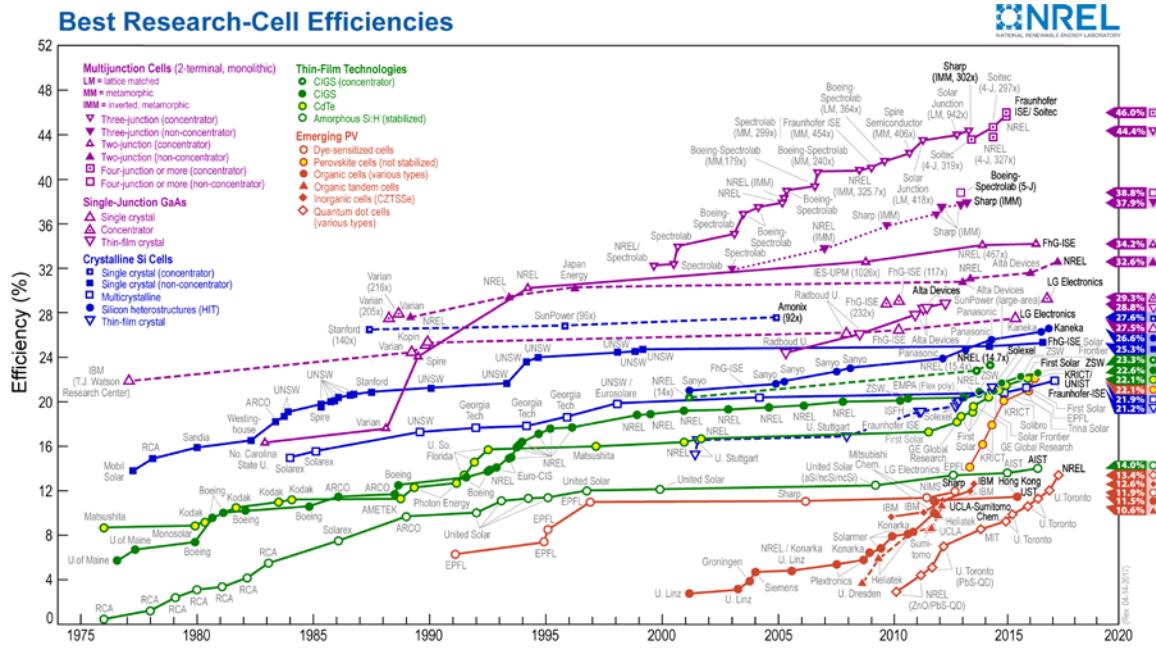


Figure 10. Efficiency Chart by Solar Cell Design.
Source: National Renewable Energy Laboratory (2017).

The high cost of multi-junction solar cells limits their usage to very specialized applications, such as space operations. The main drivers of cost include the high price of materials and technological challenges associated with manufacturing very complex structures.

3. Summary

This section introduced the major chemical, physical, and constructive characteristics of single solar cells to provide an understanding of the basic changeable parameters for influencing the efficiency of solar cell design. The section presented a promising way to achieve higher efficiency by implementing solar cell basics and developing the design into multi-junction solar cells. The goal of this thesis is to develop an efficient way to simulate various solar cell designs and obtain fast results.

This work focuses on internal solar-cell design efficiency. This thesis does not consider the following aspects of solar-cell efficiency:

- Reflection of light off the cell surface
- Temperature increases that impede the flow of charge and, therefore, a decrease in efficiency
- Internal recombinations of electrons and holes that reduce the cell's current flow ability
- Semiconductor material resistance that decreases the power output by an internal voltage drop
- Material defects that create energy-absorbing states within a material's band and decrease cell current
- Shading spots from front metal grids that reduce the amount of light reaching the cell

These areas remain subjects for future research.

B. SILVACO ATLAS SIMULATION ENVIRONMENT

Silvaco ATLAS is a one-, two-, and three-dimensional, physically based semiconductor-device simulation software, originally developed to mimic the physical implementation of structures and predict the electrical characteristics of electronic devices by SILVACO Inc (2004). This kind of simulation has become important because it is quicker and less expensive than conducting experiments to glean insights, which may be impossible due to the nature of the physical experiments. The Silvaco ATLAS simulation works by spanning a pre-defined mesh over the implemented device structure and computing the electrical characteristics for each of the mesh points. The simulation generates a result file, which requires analysis.

This research compares the work by Tsutagawa (2013), which utilized a genetic algorithm to optimize the simulation output, to the recently developed NOLH design of experiments to reveal the applicability of the new approach in the context of optimizing solar-cell designs.

1. Main Characteristics

Silvaco ATLAS uses two sets of input. The first file contains the structure description of the electronic device, and the second file contains the structure parameters for the simulation run. The objective of this thesis is to validate the NOLH design of experiment by varying the structure parameters in a predefined region on a fixed device such as a solar cell. The NOLH design is available as a Microsoft Excel spreadsheet that can generate the properly scaled design points for input. A design point is a unique input parameter combination. A novel approach implementing Silvaco ATLAS simulation software on the Naval Postgraduate School (NPS)'s high performance computer, called "Hamming," leverages the ability for parallel processing and reduces the computational time compared to the sequential genetic algorithm optimization. Unfortunately, due to license server issues the parallel processing capacity could not be implemented during the time of this research and is considered future work.

2. Emulating a Solar-Cell Structure

In a first step to optimize an electronic structure using the Silvaco ATLAS software, an electronic representation of the device needs to be implemented in the simulation software. The physical properties are specified by the kind of material, the dimensions of the material (region), the doping material, and the doping level. Simulating the device and its behavior entails computing electrical properties at predefined points (using a mesh) inside the material and placing specific measurement points in the structure to gain insights. (SILVACO Inc, 2004).

a. Mesh

The construction of a mesh within the device's structure is the essential part of simulating electrical characteristics. The mesh spans the entire structure, and at each mesh triangle, a node is created. At these nodes, Maxwell's equations are used to determine the coarseness of the mesh. The analyst has to make a tradeoff between accuracy and numerical efficiency in the calculation.

Accurate results require a fine mesh but increase computational time. However, computational efficiency requires a coarse mesh that minimizes the total number of grid points, hence reducing the simulation run time. A good mesh structure requires a dense mesh in the areas of the structure where energetic transitions happen but wider spread nodes in less active zones.

b. *Regions*

The regions are numbered areas that refer to the mesh nodes' specific information, assigning them material property parameters. An entire mesh must be assigned to a particular region.

c. *Materials*

Physical parameters of commonly used materials are defined in model libraries in the simulation environment itself, which contains among other properties the band gap, mobility, and lifetime. The more accurate the parameters, the closer the simulation result to the results of a physical cell. This applies particularly in cases for which materials are required that are not in the database. In addition to the material parameter itself, simulating solar cells requires data files on the optical properties to yield accurate simulation results.

d. *Electrodes*

Electrodes can be assigned to specific regions in the structure to gain insight into the electronic resistance conditions in those areas.

e. *Doping*

As highlighted in the semiconductor basics, doping levels are very important for manipulating the electric characteristics of semiconductors. Previously assigned regions will be doped as one of the inputs to the experimental design.

3. Summary

Silvaco ATLAS was not primarily developed for simulating solar cell-type applications. However, due to the built-in function to mimic light-sensitive or light-emitting semiconductor applications, such as photo sensors or light emitting diodes (LED), the program proved an accurate simulation tool when it validated the results gained from a research paper published by the Japan Energy Corporation's Central Research Laboratory for a dual-junction cell (Agui et al., 1998).

Compared to the existing physical process of developing solar cells through hardware experimentation, using the Silvaco ATLAS simulation environment provides a cost- and time-effective tool for optimizing solar cells.

C. OPTIMIZATION

For a better understanding of the optimization techniques used in the past and the new technique introduced in this thesis, this section provides an overview of the main features of the two implementations. It also answers the question of why NOLHs are a better way to optimize solar-cell designs in the Silvaco ATLAS simulation environment.

1. Genetic Algorithm

A genetic algorithm is a metaheuristic process to find solutions in optimization or search problems, inspired by natural selection processes such as mutation, crossover, and selection (“Genetic Algorithms Tutorial”, 2017). This procedure provides a stochastic yet systematic way to inspect the factor space and will usually gradually converge toward the best solution. The genetic algorithm is used to solve optimization problems that are not well suited to standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic or nonlinear. Considering the seven-bit implementation used by Tsutagawa in 2013, the size of 128^{10} possible combinations for a single-junction cell shows the complexity the

algorithm has to solve. The algorithm is used to provide sufficiently good solutions whenever problem information is incomplete or imperfect or the computation time is limited.

The basic mechanism used is adapted from nature, based on a theory introduced by Darwin (Darwin & Wallace, 1858). Parameters of the solution space are coded into a machine-readable format, or bit string, called a chromosome. An example of an eight-parameter solution space—with 16 design variations per parameter—of a single-junction solar cell mapped into a four-bit granularity genetic algorithm coding, used by Bates (2004), is shown in Figure 11.

Window Thickness		Window Doping		Emitter Thickness		Emitter Doping		Base Thickness		Base Doping		BSF Thickness		BSF Doping	
Genes		Genes		Genes		Genes		Genes		Genes		Genes		Genes	
0	0000	0	0000	0	0000	0	0000	0	0000	0	0000	0	0000	0	0000
1	0001	1	0001	1	0001	1	0001	1	0001	1	0001	1	0001	1	0001
2	0010	2	0010	2	0010	2	0010	2	0010	2	0010	2	0010	2	0010
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	1111	15	1111	15	1111	15	1111	15	1111	15	1111	15	1111	15	1111

The diagram illustrates the mapping of 16 genes (4x4 matrix) onto a 32-bit chromosome. The genes are organized into four columns, each representing a different parameter: Window Thickness, Window Doping, Emitter Thickness, and Emitter Doping. Each column contains 4 rows of binary values (0, 1, 2, ..., 15). Arrows point from each gene column to its corresponding bit position in the resulting 32-bit chromosome. The chromosome is represented as a horizontal sequence of 32 bits, divided into four groups of 8 bits each, corresponding to the four parameters. The bits are color-coded according to the original gene columns: Window Thickness (pink), Window Doping (orange), Emitter Thickness (green), Emitter Doping (blue), Base Thickness (purple), Base Doping (light blue), BSF Thickness (yellow), and BSF Doping (light purple).

Figure 11. Example 32-bit Chromosome Genetic Representation of a Semiconductor Application. Source: Bates (2004).

The algorithm chooses a random instance of the gene code, a chromosome, and uses this selection as the starting point, or chromosome parent. After evaluating the efficiency of the parent-chromosome simulation run, a child chromosome is generated by applying mutation (see Figure 12) and crossover (see Figure 13) techniques.

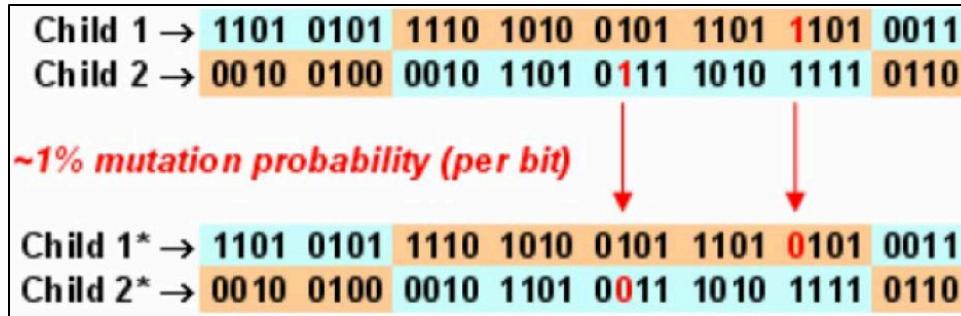


Figure 12. Example of 1% Mutation of Genes in Chromosome.
Source: Bates (2004).

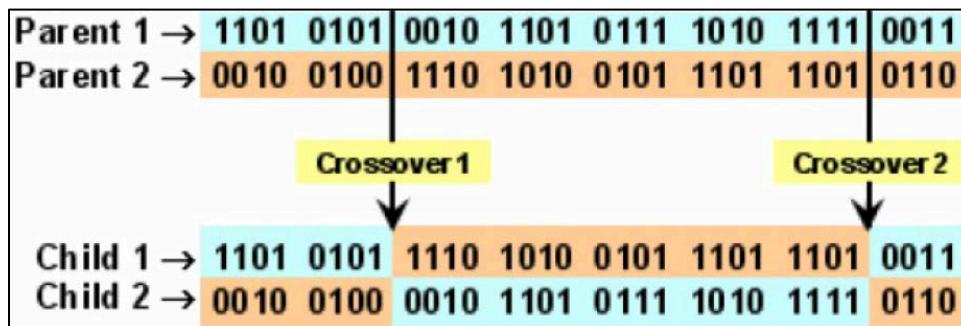


Figure 13. Example Dual-Point Crossover of Genes in Chromosome.
Source: Bates (2004).

After comparison, the simulation result from the parent and the child with the fittest gene representation is selected. If the parent instance has a better performance, it becomes the starting point for a new child instance. If the child combination produces a better result, it becomes the new parent chromosome. There is not a single genetic algorithm implementation. Since it is a broad research area, this thesis focuses on the algorithm used by Tsutagawa (2013) in his dissertation.

a. *Implementation by Tsutagawa*

Tsutagawa (2013) implemented the solar-cell parameters into binary genes and combined the genes into chromosomes to build the entire solution space of a complete cell using a seven-bit gene representation for each parameter (128 design variations per parameter). Compared to Bates who used

only four bits to represent a gene, Tsutagawa drastically increased the resolution of the experimentation space. In his research, Tsutagawa varied the breeding process using triple-point crossovers and various mutation rates (one, two, and 3 percent). Each simulation run was analyzed immediately to see whether the result yielded better power output for the solar cell than in the previous iteration. This process is called “survival of the fittest,” and if a chromosome is evaluated as fit, it is reinserted in the pool of parent chromosomes for another breeding process. In his implementation, Tsutagawa used the elitist strategy, which preserves the fittest chromosomes in a succeeding generation. This strategy increased the convergence tempo and grew a much healthier and stronger population, hence finding the solution closest to the optimum. The genetic algorithm provided a means for accomplishing a local hill-climbing search capability coupled with a random feature to jump out of local maxima due to the mutation feature.

b. Results

In his dissertation, Tsutagawa (2013) combined all previous research in optimizing multi-junction solar cells and was able to develop and refine the genetic algorithm approach to improve the solar cell design efficiency. He listed seven major improvements to Bates’ (2004) original research on genetic algorithm implementation. Tsutagawa’s genetic algorithm-related enhancements were: 1) a seven-bit gene representation, which resulted in a much denser solution space coverage, 2) different mutation rates, and 3) a changed crossover rate, which improved the performance of the genetic algorithm itself. Arguably, the improvement by using Java® (ORACLE INC, n.d.) instead of MATLAB® (The MathWorks Inc, n.d.) is not worth mentioning as the tastes of the programmer determine the programming language he uses.

Although Tsutagawa (2013) was able to improve the performance of a single-junction solar cell by 1.1% to 4.0% and 2.3% for the dual-junction solar cell depending on the substrate used, the origin of the improvement is unclear. A

higher resolution of the experimentation space and longer runtime of the genetic algorithm should perform better, but the contribution of the mutation or crossover rates in improving the results is uncertain.

A weakness of Tsutagawa's approach is the complexity of the chromosome and, hence, the runtime of the simulation—up to 102 hours. The computational time reached a level that is unsatisfactory in developing more advanced solar-cell designs. Considering the number of parameters to simulate a solar cell with four or more junctions, it would be nearly impossible for the genetic algorithm in the current implementation to optimize the problem in a reasonable amount of time.

Therefore, this thesis introduces a new approach to overcome these limitations, utilizing the NOLH design of experiments to optimize solar-cell designs. The following paragraphs highlight the basic characteristics and features of the NOLH design for the experimental Silvaco ATLAS simulation setup.

2. NOLH Design of Experiments

The NOLH is a modern design of experiments (DoE) technique. The basic idea behind the DoE is the efficient generation of values for simulation input variables, also called factors, to enable analysts to determine whether and how they affect the output, or response. After running the simulation with the input design, an analyst fits a regression model, and the resulting response surface metamodel approximates the input/output relationship. Getting an accurate fit from the metamodel requires calculating many possible points in the surface. Doing so necessitates a good space-filling property. Another important feature of the NOLH is orthogonality. A nearly orthogonal design means that effect estimates are nearly independent of each other. Orthogonality also provides minimum variance in estimating the response model coefficients. Thus, a smaller pairwise correlation between design columns is better.

In the past, researchers had the problem of compromising between minimal correlation and space-filling properties. McKay, Beckman, and Conover (1979) developed the Latin hypercube design to address the need for space-filling design with continuous factors. Follow-on research aimed to address the high correlation confounding factor interrelations and the inability to accept different input factor types (discrete, categorical and continuous). One part of the research focused on correlation by sampling designs at the corners, edges, and center of the experimental space. The downside of this method is the loss of information in the regions between the design points. Another part of the research tried to make the design flexible and accept multiple types of input factors to adapt for different simulation requirements.

The NOLH design of experiments developed by MacCalman (2013) symbolizes the state of the art of space-filling designs with minimal correlation between all second-order terms to explore complex simulation models. The property of accounting for second-order relations is its main distinction from the genetic algorithm. This behavior allows the analyst not only to identify the significant input factors but also to reveal the system's complexity by showing the interactions between input factors. Considering only linear effects reveals the input parameters that have no effect or the most effect on the output of the simulation. Including quadratic terms rather than assuming the linear model for the simulation response might increase the information from the simulation output with smaller changes of the input parameter (diminishing returns). The third area the NOLH design is able to efficiently handle is two-way-interactions or synergistic effects. Due to the nature of the NOLH design, input parameters are varied simultaneously with each simulation step instead of varying only one parameter and keeping the others constant. With this characteristic, we are able to see the influence of one input parameter on the other input factors and their relationships. The history and intermediate steps that led to the second-order NOLH design are depicted in Figure 14.

Space-Filling and Second Order Design Domain Convergence

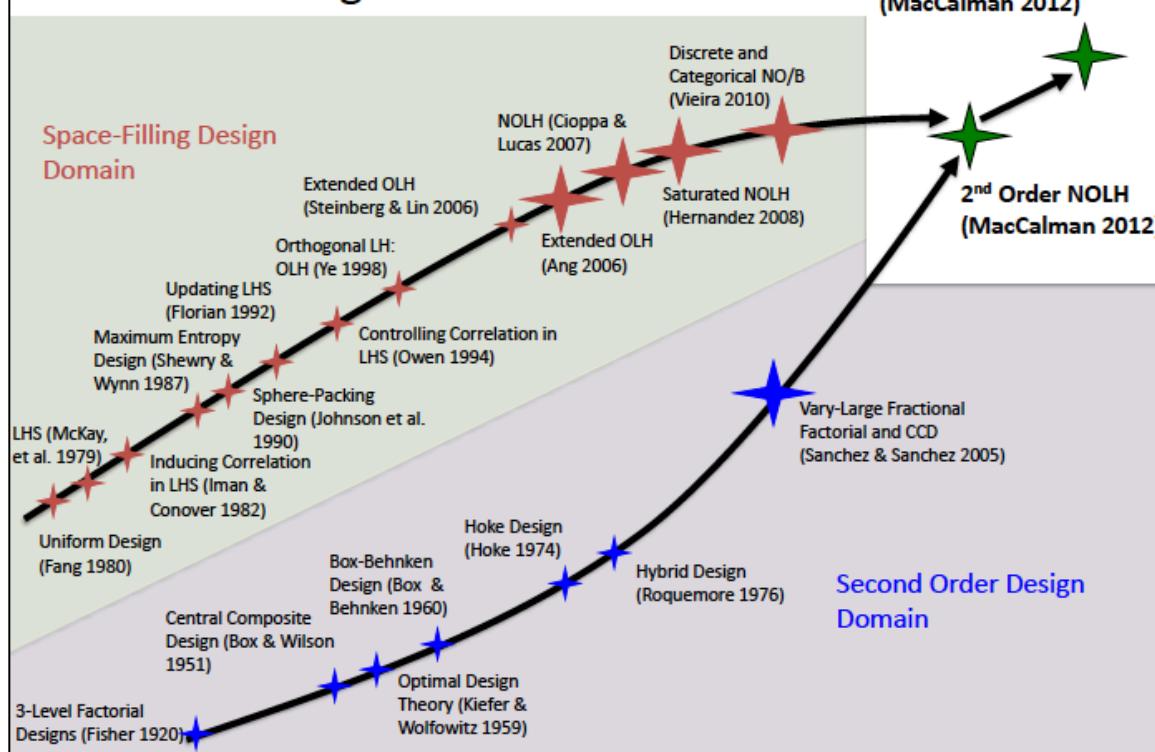


Figure 14. Research Steps Toward the NOLH DoE.
Source: MacCalman (2013).

After running the simulation and loading all the input and output parameters into a statistical analysis tool, a metamodel, which acts as a surrogate of the simulation, is computed using a stepwise regression technique. The resulting metamodel or response surface approximates the functional behavior between the simulation input parameters and the maximum output power per simulation run. With the stepwise regression, the model determines the input parameters that have the greatest effect on the output value. The metamodel also has a predictive property to conduct a sensitivity analysis of the input parameter and its effect on the output. This can be used to adjust the parameter values to get the largest output and the optimal input parameters. To represent the second-order model mathematically, a polynomial model is used covering all linear, quadratic, and two-way interactions, as shown in Figure 15

(MacCalman, 2013). With this model, we are able to explore the complex response surface of the simulation and find the local and global minima and maxima, stationary points, and ridges and saddles.

$$y = \beta_0 + \sum_{j=1}^k \beta_j X^j + \sum_{j=1}^k \beta_{jj} (X^j)^2 + \sum_{i=1}^{k-1} \sum_{j>i}^k \beta_{ij} X^i X^j + \epsilon$$

Meta-model of Simulation Model
"model of a model"
Surrogate of the Simulation

Linear Effects Non-linear Quadratic Effects Interaction Effects (Synergies) Random Noise

Figure 15. Mathematical Metamodel of NOLH DoE Simulation Result Analysis. Source: MacCalman (2013).

a. *Implementation of the NOLH*

In his dissertation, MacCalman (2013) introduced a “genetic algorithm” (GA) for generating designs that allow experimenters to simultaneously identify critical input variables and fit commonly used second-order models with nearly uncorrelated coefficient estimates.” This design is called a nearly orthogonal Latin hypercube as it produces simulation parameter inputs as a Latin hypercube with an absolute pairwise correlation between any two input parameters no greater than 0.05 (Hernandez, 2008). By not exceeding this threshold, results suffer minimal adverse multicollinearity effects. The NOLH designs also provide “flexibility to fit more complex relationships on a modest number of factors” (MacCalman, 2013). In the case of simulating solar-cell designs with Silvaco ATLAS, this property of design generation is essential since the input factors require a wide range of values. For the simulation of semiconductor material doping, levels with a range of 10^{16} to 10^{20} are necessary. The NOLH design implementation in Excel prevents generating these high numbers. In order to overcome the limitation, a transformation of the design parameter values is considered.

The NOLH-generating algorithm developed by Cioppa and Lucas (2007) was implemented in Microsoft Excel by Susan M. Sanchez (2011) and is freely available at the NPS' Simulation, Experiments, and Efficient Designs (SEED) Center website (see <http://my.nps.edu/web/seed/software-downloads>). The tool does not use licensed software or external libraries. As depicted in Table 1, creating the design of experiments with the tool provided by Sanchez was straightforward and limited only by the number of design factors. Currently, the tool is available for up to 29 parameters, which generate 257 different design points, to run in the simulation environment.

Table 1. Head Snapshot of NOLH Design Generating Matrix.

low level	1	1	1	1
high level	257	257	257	257
decimals	0	0	0	0
factor name	103	227	153	158
	31	103	216	246
	42	153	31	144

For designs with more than 29 factors, using Vieira's (2012) nearly orthogonal balanced (NOB) DoE implementation is recommended. It is also available at <http://my.nps.edu/web/seed/software-downloads>.

To create the experimental design, the user has to provide the tool with the upper and lower bounds of each factor and the number of decimal digits required, and the algorithm computes all intermediate design points (see Table 1). Once all required factors are specified in the tool, the design is ready to import into the simulation system and run. Each row of the table contains the values for the parameters and represents the input space for one simulation run. Before the start of the simulation using the NOLH, all experimental designs are generated, a major difference from the genetic algorithm.

b. Results and their analysis

Each single simulation run produces one or more output values, the response values of the simulation. In order to capture the respective result, one or more columns of the existing simulation matrix have to be created and filled with values.

Once all the simulation runs have completed, the evaluation of the entire simulation space, consisting of the input values and the simulation's output values, can start. For the evaluation, we have to build the response surface metamodel. The first step is loading the whole matrix into a statistical evaluation and model-building tool such as JMP (SAS Institute, n.d.-c). Subsequently, a model needs to be fit to capture the importance and the possible relationships of the factors.

In the case of using Silvaco ATLAS to evaluate solar-cell designs, all simulation input parameters are the x-variables, and the maximum power output of the solar-cell design is the response variable y. If a denser filling of the experimental space is required and if the computational cost allows, the design can be rotated and stacked to generate more design points.

3. Summary

This chapter highlighted the main characteristics of the genetic algorithm and NOLH along with their pros and cons to compare the two approaches. In his dissertation, Tsutagawa (2013) demonstrated the capabilities of the genetic algorithm to optimize multi-junction solar cells. He combined the Silvaco ATLAS simulation environment used by Bates (2004) with the genetic algorithm to improve solar-cell design. He was able to refine the optimization process and subsequently improve solar-cell performance by nearly four percent. Despite such improvements, this approach has exhibited some weaknesses. The genetic algorithm requires the output of the previous run to generate the next input. Therefore, the process is a closed loop and requires sequential processing. This marks one of the main disadvantages of this simulation input-preparation

technique. With more advanced solar-cell designs, simulation runs have taken as long as 100 hours to complete. Therefore, the only possibility of decreasing the runtime is to use a higher powered computer to reduce the time of each simulation run.

A more promising way is utilizing parallel processing. Several simulation runs can be computed simultaneously, limited only by the number of nodes implemented in NPS' high performance computer, Hamming. However, in the case of this research, the number of available software licenses limits the number of possible concurrent simulation runs.

To use parallel processing, the NOLH design of experiments provides the right setting. Since all simulation factor settings are pre-calculated by the NOLH, parallel runs are possible. Once the simulation inputs and outputs are stored in a matrix, statistical software can easily evaluate the results.

Parallel processing reduces the runtime proportionally to the number of processors available and enables the solar-cell developer a much faster experimentation turnaround cycle. Compared to previous experimental designs, the NOLH exhibits better space-filling properties, more design points, and better analysis properties. The reduced runtime of running the NOLH on a cluster allows more experiments in the same amount of time. This enables the developer to maintain the overall factor ranges in the input matrix but increase the resolution of the factors. To achieve this characteristic for the genetic algorithm, the only possibility is to increase the number of bits used to code each single parameter, with the downside of increasing the complexity and simulation runtime.

Another disadvantage of the genetic algorithm is that it does not give any directional information about where improvement of the design can be found. The algorithm does not exploit the shape of the response surface and as such does not support the solar cell developer to give information on further steps in design improvements.

This thesis introduces the novel use of the NOLH to design solar cells using the Silvaco ATLAS simulation environment. The implementation and results are discussed in the following chapters.

III. EXPERIMENTAL DESIGN

The focus of this research is to prove the applicability of the nearly orthogonal Latin hypercube (NOLH) in optimizing solar-cell designs in conjunction with the Silvaco ATLAS simulation environment. This chapter uses the physical model containing the cell design and performance results published by Agui et al. (1998) and verified by Tsutagawa (2013) to replace the genetic algorithm and compare the accuracy and validity of the experimentation results. The second goal of this chapter, besides determining general applicability, is developing an easy process for creating the design of experiments (DoE), running the design in Silvaco ATLAS, and analyzing the results. This keeps the focus of the solar-cell developer on the solar-cell design itself without getting distracted by the very complex process of generating the optimal cell design.

A. EXPERIMENTAL SET UP

To utilize the full advantage of the NOLH DoE, parallel computing is highly desired. At NPS, the high-performance computer cluster Hamming is the first choice in that all setup issues can be handled locally. In the case of this research, it was essential since the Silvaco ATLAS simulation system requires a very specific set of software to run in a distributed environment. License handling with the server was a challenge in making this happen.

Since the setup of Silvaco ATLAS on Hamming could not be finished prior to the start of the experimentation phase, the sequential approach was used at the beginning. The advantage of starting this way was that initial results were being generated and the process to evaluate the simulation output could be developed further. Moreover, a direct comparison of the runtimes of both approaches was possible under the same experimental and cell design setup.

1. Framework / Assumptions

The general experimental setup is shown in Figure 16. The starting point is the solar-cell design to be simulated and the resulting factors that represent the design within the simulation environment. Then, the factors and their ranges are specified within the NOLH Excel spreadsheet. Once this is done, the resulting matrix of factor values are transferred to files in the Silvaco ATLAS simulation environment, which is now able to execute and produce the deterministic simulation result. Since each factor matrix row represents the value set for one single simulation run, a software script has to make sure each set runs and the simulation result is captured. The software script also ensures the logarithmic transformation of the doping levels since the NOLH spreadsheet does linear scaling and cannot handle the values required in the 10^{16} to 10^{20} range. Once the simulation has completed, the design matrix and the appended outcome column are loaded into the statistical analysis software and analyzed. The resulting metamodel gives an estimate of the factor values that optimize the solar-cell design; that is, they produce the most power. These analysis steps mark the end of one optimization iteration and can be repeated as appropriate.

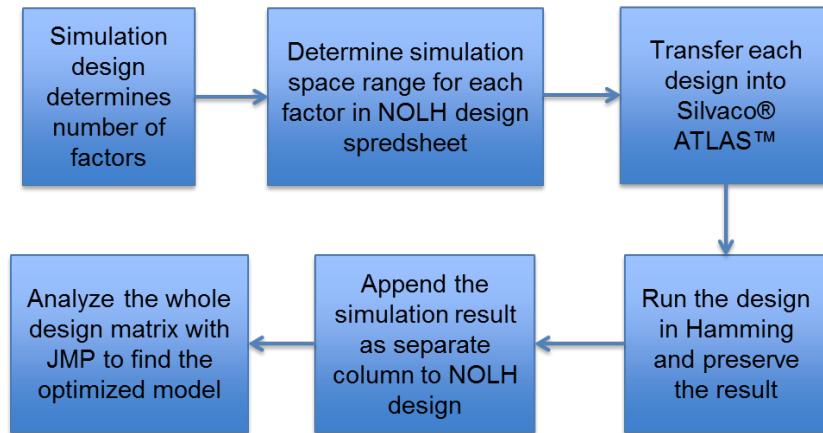


Figure 16. Experimentation Setup Schematic.

The aforementioned experimentation setup can be repeated as required to compress the factor space for each single variable and to increase the resolution

in the whole factor space. The repetitive experimentation is one of the achievements of the NOHL over the genetic algorithm, made possible only by the simple generation procedure for space parameters using the NOLH Excel spreadsheet and the runtime reduction utilizing parallel computing with Hamming. These factors focus on the design of the solar cell rather than spend time designing the optimization algorithm for the experiment. Analyzing previous work in this field highlights that 80% of the time was invested for the optimization algorithm itself and 20% for the solar-cell design. One of the main achievements of this research is that the developer of solar cells now has an easy-to-handle tool for focusing on solar-cell design rather than spending time on optimization.

2. Design Creation

The basis for creating the experimental design is the NOLH Excel spreadsheet developed at the NPS SEED center. Taking the Agui et al. (1998) design as the baseline, the published values for thickness and doping levels have been taken and lower and upper bounds placed in the heading section of the spreadsheet. The Agui design emulated in Silvaco ATLAS requires 24 different factors to characterize all imported design features.

Table 2. NOLH Excel Header with Some Example Values.

low level	0,01	17	0,02	17	0,2	16
high level	0,05	19	0,08	19	0,8	18
decimals	3	3	3	3	3	3
factor name	winthick	windopconc	emthick	emdopconc	basethick	basedopconc
	0,026	18,766	0,056	18,227	0,577	17,797
	0,015	17,797	0,07	18,914	0,631	17,063
	0,016	18,188	0,027	18,117	0,777	17,625
	0,026	17,32	0,044	18,266	0,638	17,492
	0,025	18,117	0,057	17,813	0,758	17,586
	0,028	17,734	0,077	17,32	0,718	17,813

An extract of the design parameters is shown in Table 2. Notably, the values for the doping level reflect only the exponent, and in a follow-on step, these values have to be transformed into the region the simulation system

requires for the values. This is done with a 10-to-the-power of the design value transformation. Once the intermediate values have been computed by the spreadsheet, only the design points have to be copied and pasted into a different Excel spreadsheet. Finally, the new document with the design values is saved as a comma-separated values (CSV) format file in the folder where the experimentation runs later.

To improve the coverage of the experimentation space, a technique called rotating and stacking is applied. First, the header of the NOLH Excel with the factor's lower and upper bonds is selected and copied. Then, the entire copied section is pasted one column to the right (a rotation). The resulting intermediate values generated by the spreadsheet are different from the ones generated in the previous step. Next, they are copied and pasted again beneath the previous generated values in the CSV document. This procedure can be repeated as required, keeping in mind with each rotation and stacking the number of experiments increases in our case by 257, which at the end requires a longer runtime. The analyst must find a balance between the experimentation space density and coverage and the simulation system runtime.

3. Silvaco ATLAS Input File Setup

The NOLH design point insertion into the Silvaco ATLAS solar-cell design file is the next step in the sequence. The solar-cell design parameters in the existing design template file have to be overwritten with the values generated in the NOLH Excel and saved in the CSV file with a unique name to ensure the right mapping of design input and output. In addition, all the output statements forcing Silvaco ATLAS to generate an output file have to be numbered as well to ensure the output data can be mapped to the generating simulation run.

There are two approaches to tackle this step in the process. One is to generate as many design files as rows in the CSV matrix (all in one); the other is to do it sequentially as required by the program controlling the simulation runs in Silvaco ATLAS.

In this research and in the attempt to utilize Hamming, the all-in-one approach was used, and the respective number of experiments generated simulation files with numbered naming for each file.

The practical application is a Python script (Python Software Foundation, n.d.), which assigns the NOLH-generated values, with the parameter variable in the Silvaco ATLAS solar-cell design template. It is worth mentioning that for this step, having the solar cell-defining variables in one single block one after another reduced the complexity of assigning the NOLH values to the right Silvaco ATLAS file parameter variables using the Python script.

B. EXPERIMENTAL EXECUTION

1. Silvaco ATLAS Simulation Run

Due to the unavailability of Hamming, the initial experimentation was done sequentially. A Python script was run to preprocess the input files. A second Python script controlled the simulation runs in Silvaco ATLAS and captured the runtime for each simulation for evaluation purposes.

Utilizing Hamming would not have required such a control mechanism. The system itself would have picked the simulation file to run and saved the output for further analysis.

2. Simulation Output

Each simulation run generated a number of values for the cathode current versus anode voltage curve under the AM0 spectrum of light, as shown in Figure 17, and was saved in a text file format.

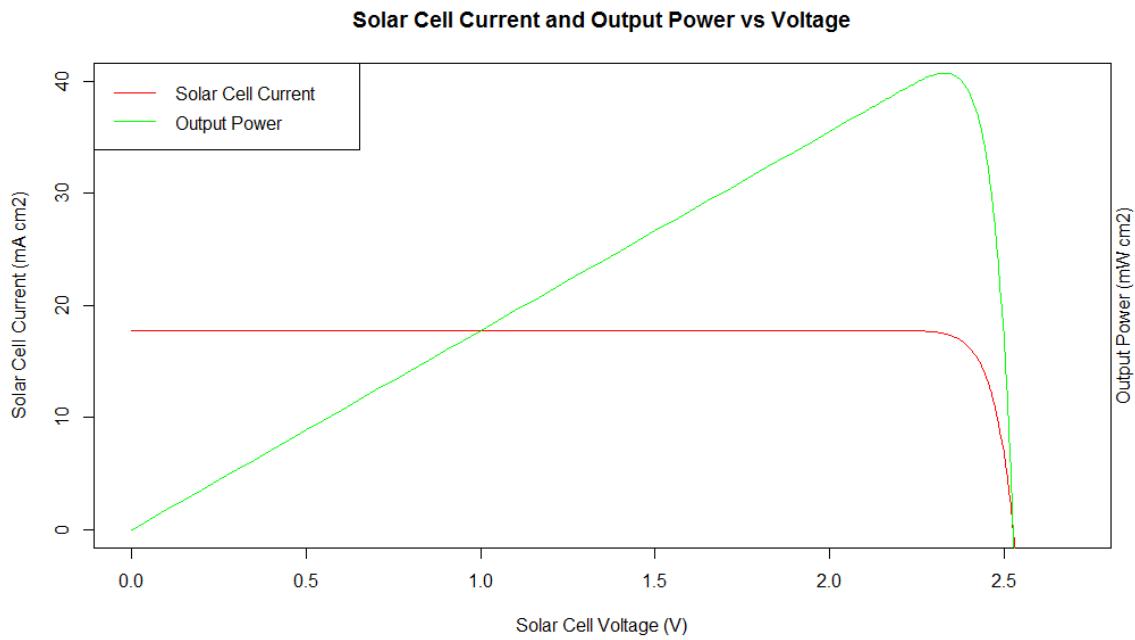


Figure 17. Current vs. Voltage (I–V Curves) and Output Power vs. Voltage Generated from the Maximum Output Power Simulation Run.

For analysis purposes, all output files had to be read and processed with a Python script, and the solar cell's current and voltage values provided by Silvaco had to be multiplied to calculate the output power. The maximum overall power value for each simulation run was saved in a vector, which appears as the green curve in Figure 17.

Once all simulation runs finished, the vector with the maximum power values was appended to the solar-cell design input CSV file. In the same processing step all doping level transformation values were calculated and written in the input and output CSV file as the final step to prepare the data for analysis. In this research, JMP by SAS was used to analyze the data.

Another preparation step, namely copying the names of the parameter in a headline and saving the CSV file under a different name before analysis, proved to be helpful. This made the analysis more “readable”.

IV. EXPERIMENTAL ANALYSIS

The starting point for analysis of the simulation is loading the CSV file, which contains the simulation input values and the maximum output power values, into the analysis software, JMP. The general sequence to analyze the data is as follows:

1. Get a general overview of the data, which includes checking for unusual observations and the general dispersion of each single parameter or outliers. Find the factor combination that generates the highest output power values among all experiments. In the case of this experimental setup, find the experiments whereby the simulation environment could not converge to a result and gave a zero or close-to-zero output power result.
2. Fit stepwise a multiple regression model to analyze the input and output relationship. The fitted model aids in understanding the relationship between the response variable and the factors, including main effects, interactions, and non-linear effects. The result of the prediction profiler indicates the direction where to shift the factor range in the next iteration.

A. EXPERIMENTAL ANALYSIS

Within this research, five different sets of NOLH designs were used to simulate the Agui solar-cell design. In a first attempt, two basic 257-design-point NOLH designs were utilized with the goals of proving the general applicability of the concept and producing initial simulation results. The difference of the designs was the method of transforming the doping levels.

Subsequently, the technique of rotating and stacking was applied to improve space-filling in the designs and improve the resolution to explore the experimentation space. By fully utilizing the capacity of the NOLH-generating Excel spreadsheet, five rotations and stackings produced two 1,542-design-point

designs for which only the linear-transformed doping-level design was analyzed. Later, two iterations of an improved 1,542-design-point design were simulated, with the focus of improving the output power by narrowing the experimentation space to the most promising regions for higher granularity.

The aforementioned two-step sequence was used to analyze the data, reveal insight into the input/output relationship, and better understand the interactions among the design factors and using the locality information to improve the design.

1. Agui Solar Cell with 257 Design Points

Considering the genetic algorithm method previously used to find the optimal solar-cell design, the research could have stopped at this point because the experimentation found a cell configuration that produced the maximum power output among all input value combinations. One row of the NOLH design's input produced an output power of 40.76 mW/cm^2 . From an engineering standpoint, an increase of more than 10% may have been good enough, but no insight on the data would have been gleaned. The following paragraphs show some steps in statistical analysis and the interpretation of the results.

a. General Data Screening

Data analysis typically starts with a screening of the data to detect anomalies within the dataset. For the first 257-design-point logarithmic-transformed Agui solar-cell design, 13 out of the 257 simulation runs produced a zero output value. Those experiments were excluded from further analysis. The dispersion of some of the variable design points is depicted in Figure 18. As shown, not all the areas are equally dense due to the exponential transformation.

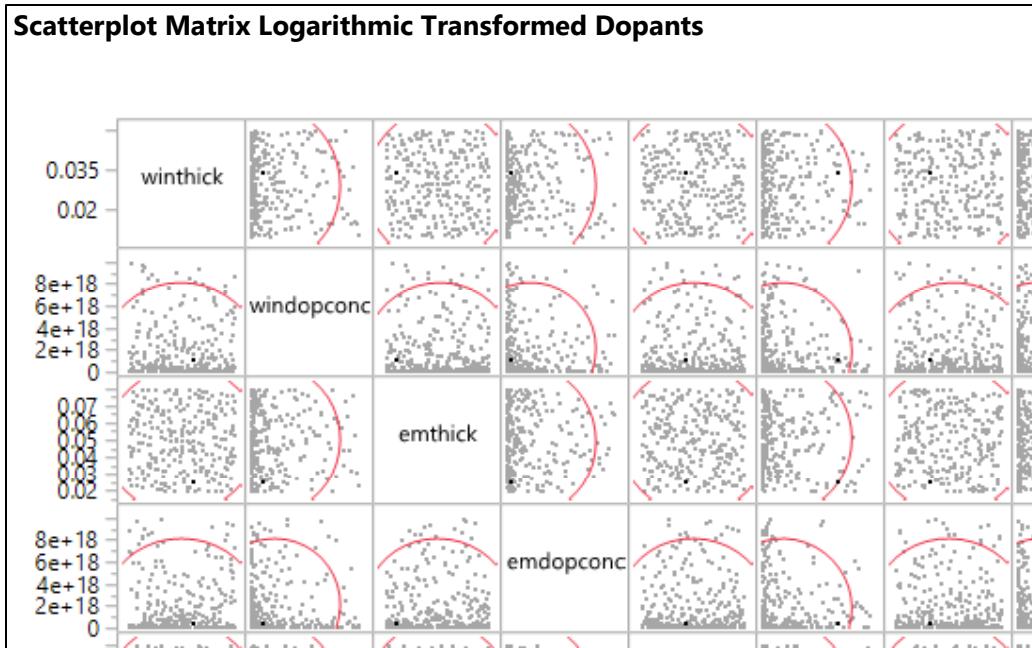


Figure 18. Extract of Scatterplot Matrix of 257-Design-Point Agui Cell Design Using Logarithmic-Transformed Dopants.

The semiconductor's thickness-layer parameters are distributed all over the space, with some limitations at the edges and corners, while the values for the doping level simulation are denser at the lower side of the area. This is due to the logarithmic transformation and the effect of small differences of the pre-transformation numbers resulting in huge differences after transformation. The dissimilar distribution of the logarithm is the reason that only the linear-transformed doping levels were considered in further experimentation.

An extract of linear transformed doping input values is depicted in Figure 19. As shown, the linear transformation covers the whole factor space much better than the logarithmic-transformed values do.

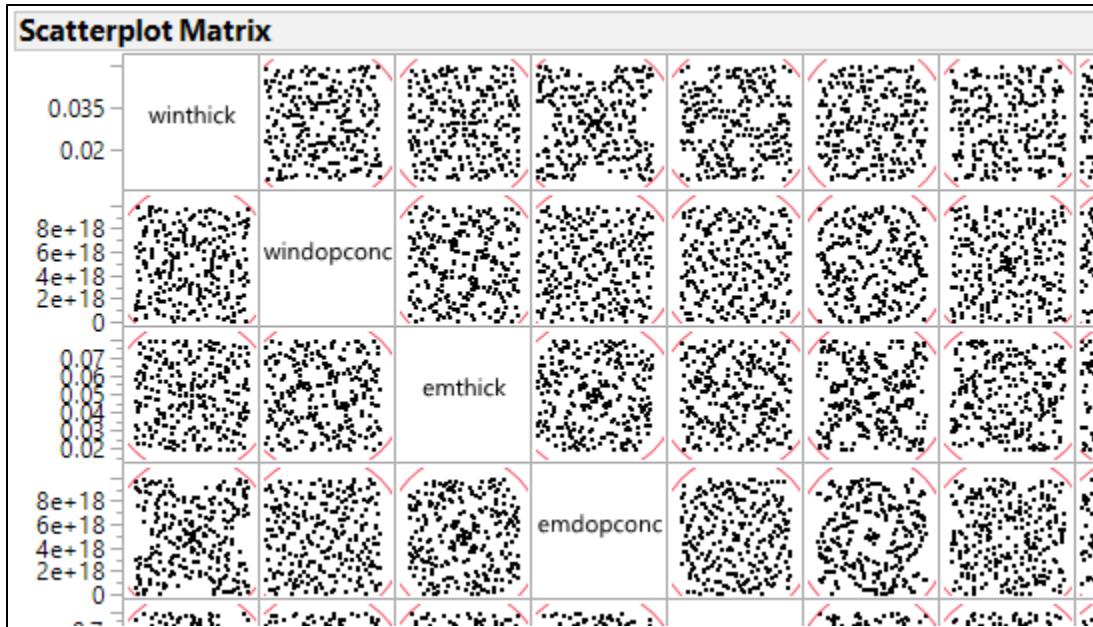


Figure 19. Extract of Scatterplot Matrix of 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

The matrix in Table 3 shows all of the correlations among the input parameters and the simulation result values. As constructed, the NOLH design kept the correlations among the input variables as small as possible; hence, results suffered minimal adverse multicollinearity effects.

Scanning through the values shows the correlations between the input variables are below 0.05, as required by Hernandez (2008). The output variable gives a clear indication of which input variables have a significant effect on the simulation output. The fitted regression model in the following section supports this assertion.

Table 3. Correlation Matrix of Input Parameters and Output Power Values of 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants

Correlations	winthick	windopconc	emthick	emdopconc	basesthick	basedopconc	bsfthick	bsfdopconc	buffthick	bufffdopconc	tunemthick	tunedopconc	tunbasethick	tunbasedopconc	botwinthick	botwindopconc	botemthick	botmdopconc	botbasesthick	botbasedopconc	botbsfthick	botbsfdopconc	botbuffthick	botbufffdopconc	Sim Result
winthick	1.0000	-0.0097	0.0030	0.0231	0.0220	0.0114	-0.0094	0.0053	-0.0176	0.0138	0.0121	0.0406	-0.0064	-0.0253	-0.0082	0.0095	0.0066	0.0107	-0.0062	-0.0108	-0.0091	0.0234	-0.0090	-0.0017	-0.3271
windopconc	-0.0097	1.0000	0.0023	0.0115	0.0027	0.0225	-0.0030	0.0077	0.0045	0.0145	0.0167	0.0209	-0.0001	-0.0013	0.0037	0.0090	0.0034	0.0035	0.0109	-0.0106	-0.0112	-0.0056	0.0048	-0.0128	0.0819
emthick	0.0030	0.0023	1.0000	-0.0122	0.0225	-0.0031	0.0017	-0.0159	-0.0145	-0.0014	0.0085	0.0055	0.0061	0.0177	0.0101	0.0031	0.0016	-0.0002	-0.0149	-0.0156	-0.0189	0.0063	0.0047	0.0027	-0.0627
emdopconc	0.0231	0.0115	-0.0122	1.0000	-0.0041	-0.0119	0.0075	-0.0166	0.0074	-0.0117	-0.0028	-0.0269	-0.0058	0.0230	0.0059	-0.0070	-0.0084	-0.0011	-0.0049	0.0033	0.0032	-0.0039	0.0086	0.0114	-0.2008
basesthick	0.0220	0.0027	0.0225	-0.0041	1.0000	-0.0097	0.0075	0.0045	0.0311	-0.0018	-0.0065	-0.198	-0.0061	-0.0014	-0.0026	0.0094	-0.0173	-0.0045	0.0111	0.0356	0.0133	-0.0192	0.0037	0.0004	-0.4081
basedopconc	0.0114	0.0225	-0.0031	-0.0119	-0.0097	1.0000	0.0204	-0.0174	-0.0088	-0.0224	-0.0239	-0.0232	0.0068	0.0033	-0.0171	-0.0071	-0.0140	0.0124	-0.0179	0.0287	0.0065	0.0044	-0.0230	0.0180	0.0062
bsfthick	-0.0094	-0.0093	0.0017	0.0107	0.0075	0.0200	1.0000	-0.0090	0.0007	0.0049	-0.0013	0.0109	-0.0061	-0.0118	0.0132	0.0003	0.0115	-0.0231	-0.0065	-0.0096	0.0012	0.0051	0.0077	-0.0074	-0.0531
bsfdopconc	0.0053	0.0077	-0.0159	-0.0166	0.0045	-0.0174	-0.0090	1.0000	0.0096	-0.0037	-0.0173	-0.0230	-0.0039	-0.0027	0.0019	-0.0089	-0.0169	0.0025	-0.0299	0.0168	0.0099	0.0077	0.0090	0.0150	0.0059
buffthick	-0.0176	0.0045	-0.0145	0.0074	0.0311	-0.0088	0.0007	1.0000	-0.0206	0.0319	0.0039	0.0220	-0.0063	0.0095	-0.0039	0.0385	-0.0009	-0.0253	-0.0094	-0.0019	0.0062	-0.0195	-0.0123	0.0345	
bufffdopconc	0.0138	0.0145	-0.0014	-0.0117	-0.0018	-0.0224	0.0049	-0.0037	1.0000	-0.0071	-0.0273	0.0108	0.0112	0.0021	-0.0159	0.0133	-0.0051	-0.0124	0.0083	0.0107	-0.0070	-0.0150	0.0038	0.1936	
tunemthick	0.0121	0.0167	0.0085	-0.0028	-0.0065	-0.0239	-0.0013	-0.0173	0.0319	-0.0071	1.0000	0.0027	-0.0321	-0.0155	-0.0097	-0.193	-0.0098	0.0020	0.0062	0.0179	0.0132	0.0021	-0.0114	0.0243	-0.1996
tunedopconc	0.0406	0.0029	0.0055	-0.0269	-0.0198	-0.0232	0.0109	-0.0230	0.0039	-0.0273	0.0027	1.0000	0.0177	0.0132	0.0130	-0.0209	-0.0179	-0.0133	-0.0118	0.0293	0.0316	-0.0144	0.0046	0.0057	-0.893
tunbasesthick	-0.0064	-0.0001	0.0061	-0.0058	-0.0061	0.0068	-0.0061	-0.0039	0.0220	0.0108	-0.0321	0.0177	1.0000	-0.0146	-0.0122	0.0224	-0.0118	-0.0141	0.0124	0.0126	0.0013	0.0167	0.0035	0.0151	-0.0047
tunbasedopconc	-0.0253	-0.0013	0.0177	0.0230	-0.0014	0.0033	-0.0118	-0.0027	-0.0063	0.0112	-0.0155	0.0132	-0.0146	1.0000	-0.0106	0.0084	-0.0039	-0.0057	-0.0128	0.0203	0.0136	0.0158	-0.0008	0.0015	0.0091
botwinthick	-0.0082	0.0037	0.0101	0.0059	-0.0026	-0.0171	0.0132	0.0019	0.0095	0.0021	-0.0097	0.0130	-0.0122	-0.0106	1.0000	-0.0019	-0.0158	0.0133	0.0077	0.0132	0.0005	0.0125	-0.0134	0.0148	-0.0903
botwindopconc	0.0095	0.0090	0.0031	-0.0070	0.0094	-0.0071	0.0003	-0.0089	-0.0039	-0.0159	-0.0193	-0.0209	0.0224	0.0084	-0.0019	1.0000	0.0064	-0.0177	-0.0102	0.0266	-0.0091	-0.0098	-0.0075	-0.0054	
botemthick	0.006	0.0034	0.0016	-0.0064	-0.0173	-0.0140	0.0115	-0.0169	0.0385	0.0133	-0.0098	-0.0179	-0.0118	-0.0039	-0.0158	1.0000	0.0122	0.0033	0.0197	0.0071	0.0031	0.0133	0.0230	-0.0276	
botmdopconc	0.0107	0.0035	-0.0002	-0.0011	-0.0045	0.0124	-0.0231	0.0025	-0.0009	-0.0051	0.0020	-0.0133	-0.0141	-0.0057	0.0133	-0.0177	0.0122	1.0000	0.0034	-0.0051	0.0135	-0.0043	0.0057	-0.0177	-0.1226
botbasesthick	-0.006	0.0109	-0.0149	-0.0049	0.0111	-0.0174	-0.0065	-0.0299	-0.0253	-0.0124	0.0062	-0.0118	0.0124	-0.0128	0.0077	-0.0010	0.0033	0.0034	1.0000	0.0186	0.0067	0.0052	-0.0059	-0.0005	0.3387
botbasedopconc	-0.0106	-0.0106	-0.0156	0.0033	0.0356	0.0287	-0.0088	0.0168	-0.0094	0.0083	0.0179	0.0293	0.0126	0.0203	0.0132	-0.0102	0.0197	-0.0051	0.0186	1.0000	-0.0239	0.0053	0.0074	-0.0101	-0.2863
botbsfthick	-0.0091	-0.0112	-0.0189	0.0032	0.0133	0.0065	0.0012	0.0099	-0.0010	0.0107	0.0132	0.0316	0.0013	0.0136	0.0005	0.0266	0.0071	0.0135	0.0067	-0.0239	1.0000	0.0045	0.0029	-0.0006	0.6013
botbuffthick	0.0234	-0.0056	0.0063	-0.0039	-0.0192	0.0044	0.0051	0.0077	0.0062	-0.0070	0.0021	-0.0144	0.0167	0.0158	0.0125	-0.0091	0.0031	-0.0043	0.0052	0.0053	0.0045	1.0000	0.0042	-0.0136	-0.0176
botbufffdopconc	-0.0090	0.0048	0.0047	0.0086	0.0037	-0.0230	0.0077	0.0090	-0.0195	-0.0150	-0.0114	0.0046	0.0035	-0.0008	-0.0134	-0.0098	0.0133	0.0057	-0.0059	0.0074	0.0029	0.0042	1.0000	-0.0028	
botbufffdopconc	-0.0017	-0.0128	0.0027	0.0114	0.0004	0.0180	-0.0074	0.0150	-0.0128	0.0036	0.0243	0.0057	0.0151	0.0015	0.0148	-0.0075	0.0230	-0.0177	-0.0005	-0.0106	-0.0008	-0.0136	-0.0028	1.0000	0.0033
Sim Result	-0.3271	0.0819	-0.0627	-0.2008	-0.4081	0.0062	-0.0531	0.0059	0.0345	0.1936	-0.0196	-0.0893	-0.0047	0.0091	-0.0903	-0.0054	-0.0276	-0.1226	0.3387	-0.2863	0.0613	-0.0176	0.0250	0.0033	1.0000

Figure 20 shows the simulation power-output distribution and related statistics summary. Considering the 247 successful simulation runs, the histogram looks relatively normal. This is an indicator that the experimentation space yielded a wide variety of output values. The interesting area for this research is the right side of the histogram where the highest output values are found.

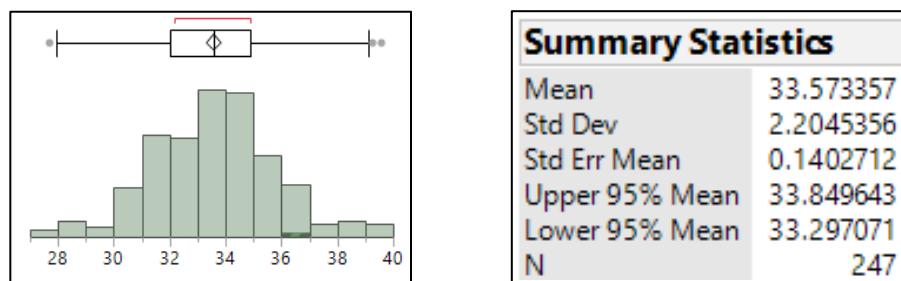


Figure 20. Simulation Output Power Distribution of the 257-Design-Point Agui Cell Using Linear-Transformed Dopants.

In general, the initial data screening revealed no major obstacles or concerns for using the data in further analysis. Better space filling is achieved through rotation and stacking. The results and their differences are discussed in the following paragraphs.

b. Model Fit and Response Surface Analysis

Given no major inconsistency in the data, a regression model was fit to characterize the effect of variation of the independent explanatory variables on the response, or output variable, using an estimated regression function called the response surface.

Due to the use of the NOLH design algorithm, a non-linear model fit was possible. In the case of this research, the JMP software (SAS Institute, n.d.-c) was utilized to fit the model and subsequently analyze the resulting response surface.

JMP offers a handy user interface to fit the model with all desired effects. This study used all main effects, two-way interactions, and quadratic terms for the explanatory variables.

Using the stepwise regression functionality, a model was fit using least squares estimation, which facilitated a search-and-select mechanism among many potential models. This approach selects a subset of effects for a regression model that provide a good model fit and improves predictive performance by reducing the variance from unnecessary terms (SAS Institute, n.d.-b).

The resulting model using the JMP's stepwise regression function is depicted in Table 4. Scanning the parameters reveals all kinds of explanatory variables and their effects in the model. Not only are main effects significant; two-way interactions and quadratic terms are also important in the model. This is one of the significant achievements using the NOLH DoE instead of the genetic algorithm; interaction effects had not been revealed previously. The NOLH approach provides a much deeper understanding of the effects input variables have on the output.

Table 4. Effect Summary Model Fit for 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Source	LogWorth	PValue
basethick	35.519	0.00000
botbasethick	28.780	0.00000
winthick	25.119	0.00000
botbasedopconc	20.063	0.00000
basethick*basethick	18.982	0.00000
botbasethick*botbasethick	17.924	0.00000
basethick*botbasedopconc	13.119	0.00000
buffdopconc	11.882	0.00000
winthick*basethick	10.314	0.00000
emdopconc	9.537	0.00000
emdopconc*basethick	5.775	0.00000
basethick*botbasethick	5.550	0.00000
botemdopconc	5.230	0.00001
botbasethick*botbasedopconc	4.867	0.00001
buffdopconc*buffdopconc	3.343	0.00045
botwinthick	2.981	0.00105
windopconc	2.265	0.00544
bsfthick	1.774	0.01681

It is worth noticing the corresponding information given by the correlation matrix in Table 3 and the effect summary in Table 4. The parameters for botbasethick and winthick were noticeable as the highest absolute values in the correlation matrix and were selected by the stepwise regression function as the most significant.

Translating these findings in the solar cell-developing world, the higher the LogWorth value shown in Table 4, the more influence the variable has on the performance of the solar-cell design. Small changes in those variables have a significant effect on the output power performance. In other words, the parameters that have been omitted from the model are not as relevant for the output power performance and, therefore, can be kept constant at economical settings within the ranges specified in the NOLH Excel spreadsheet for the experimental design.

How well the model fits is revealed in the measure of RSquare, as shown in Table 5. This coefficient of determination indicates that for the previously fitted model, 83.2% of the variability is explained by the regression. A higher RSquare value is generally better, but the algorithm implemented in JMP balances between under and overfitting the model. Overfitting would be indicated by a large difference between RSquare and RSquare Adj. An overfitted model would have a higher RSquare value but would introduce effects in the response surface that are not realistic in describing the input/output relationship of the simulation.

Table 5. Summary of Fit Model of 257-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Summary of Fit	
RSquare	0.832156
RSquare Adj	0.818906
Root Mean Square Error	0.938144
Mean of Response	33.57336
Observations (or Sum Wgts)	247

The reason for fitting a model was to use the metamodel for predicting output by varying the input variables. In the case of this research, a maximum output power of the simulated solar-cell design was desired. JMP has a built-in prediction profiler used for sensitivity analysis and desirability profiling to find the optimum response value. The technique is based on “setting up desirability functions, and searching for factor values that optimize a composite desirability of a number of responses” (SAS Institute, n.d.-a).

Figure 21 shows the JMP prediction profiler with the desirability profiling function. According to the metamodel, the tested solar-cell design should have an expected maximum output power of 40.5 mW/cm² with a confidence interval from 39.6 mW/cm² to 41.4 mW/cm².

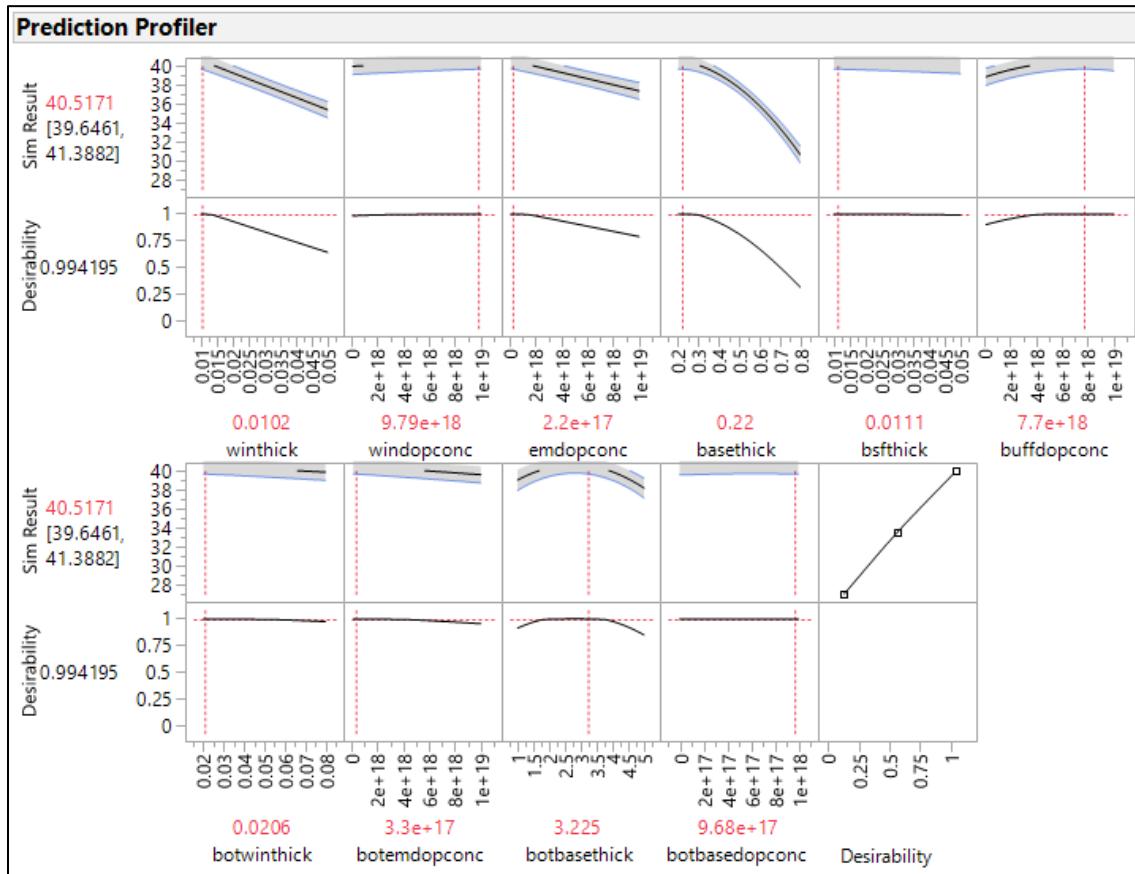


Figure 21. Prediction Profiler for Response Surface of 257-Design-Point Agui Cell Design Optimization.

The selected design factors were tested in a single simulation run to verify the values. The prediction was not exact, but using the recommended values in the simulation yielded an output power of 40.7 mW/cm^2 , which is a dramatic improvement from the basic configuration of 35.76 mW/cm^2 . The former process proved not to be as successful as anticipated. The resulting metamodel response surface optimization values from the prediction profiler tend to produce lower values as predicted. Therefore, the analysis of the experimentation was changed to using the simulation run that produced the highest outcome, and the prediction profiler is giving the direction to change the factor ranges for the next iteration.

c. Conclusion

With the process of applying the NOLH DoE in the Silvaco ATLAS simulation environment, initial results were generated and a statistical analysis of those results conducted. Already, the initial testing proved the general applicability of this approach and produced insights into the relationships and interactions of the input variables for the simulation.

Since the starting point for this research was the genetic algorithm in conjunction with Silvaco ATLAS, the following improvements may be achieved. By scanning the experimentation space, a higher maximum output power generated by a certain input parameter combination may be found. That was the achievement of the genetic algorithm application as well, but neither further insights nor parallel processing capabilities were possible.

With the statistical analysis of the results, a deeper understanding of the importance of each individual solar-cell design parameter may be highlighted. This knowledge gives the solar-cell developer a powerful tool for focusing attention and resources toward the most productive design parameters. This enhances the ability, in conjunction with the parallel processing, to achieve much faster and more goal-oriented turnarounds in the design development, which in the end should lead to a better product in a shorter amount of time.

In sum, having already promising results with the initial data run, improving the design process with more design points or more in-depth adaptation of the design-space range should lead quickly to better solar-cell designs. Following the path of the DoE in this regard yields noticeable improvements over the genetic algorithm technique used in the past. In the following paragraphs, the same solar-cell design, only with more design points, is discussed.

2. Agui Solar Cell with 1,542 Design Points (Rotated and Stacked)

For the generation of the 1,542-design-point Agui solar cell, the method of rotating and stacking was used. The goal was to get a denser filling of the design space to find a better combination of input parameters with higher output power. It was also expected that prediction accuracy of the fitted regression model would improve. The results are compared and contrasted in the following paragraphs.

a. General Data Screening and Analysis

Comparing Figure 19 with Figure 22 provides a good indication of how well the process of rotating and stacking works. With the rotated and stacked design, most of the areas in the experimentation space are densely covered, which ensures a better approximation of the input/output relationship in the regression model fit. Again, the challenge is always in finding a balance between necessary accuracy and the required runtime. In the case of this research, by using the rotated and stacked design, the runtime increased from seven to 48 hours. If this is acceptable, more experiments will improve the odds of precisely fitting the metamodel and finding an optimum.

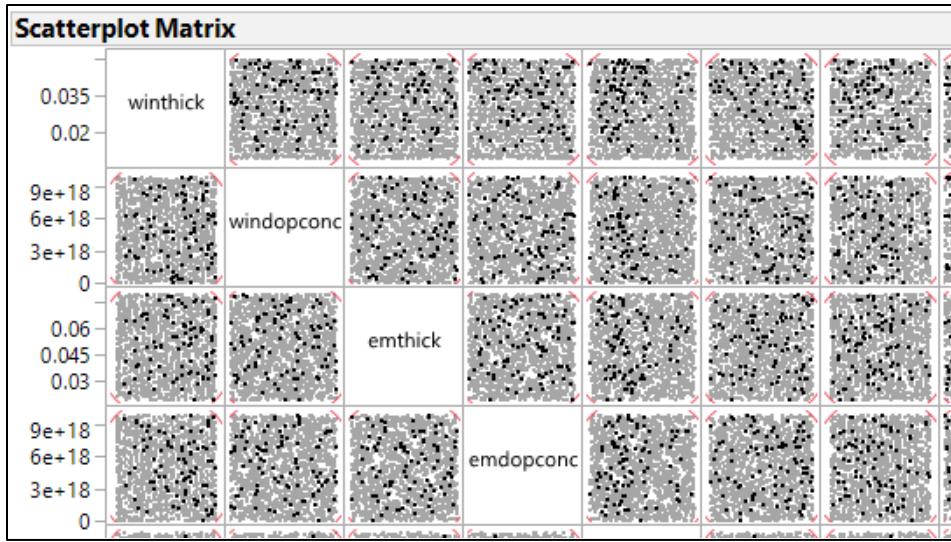


Figure 22. Extract of Scatterplot Matrix of 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Regarding output value statistics, Figure 23 shows that there was not much of a difference in the mean of the output power. The value increased slightly from 33.57 mW/cm^2 to 33.62 mW/cm^2 with a confidence interval of 33.5 mW/cm^2 to 33.74 mW/cm^2 , and this was an indicator that the additional simulation runs did not produce output with a tendency toward higher values. The factor region is still the same as in the previous experiment. The increased number of simulation runs produced output with a similar distribution, but the number of successful simulation runs is worth mentioning.

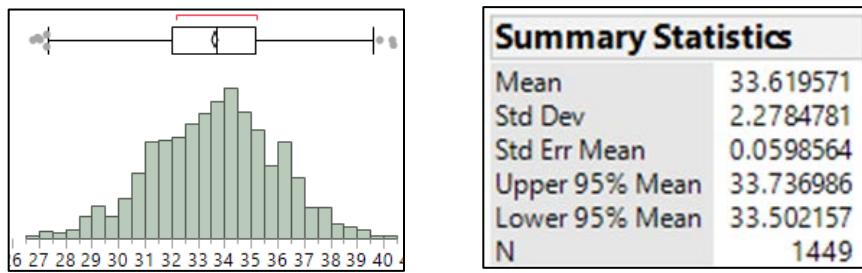


Figure 23. Simulation Output Power Distribution Improved in 1,542-Design-Point Agui Cell Design Using Linear Transformed Dopants.

The experimentation had the highest percentage of runs that did not converge. Ninety-seven out of 1,542 is a rate of 6.3%, which is approximately two percent over the rate of the other experiments.

In general, the data structure was similar to that of the basic design, with the advantage of having much denser space coverage.

b. *Model Fit and Response Surface Analysis*

The resulting model created using JMP's stepwise regression function is depicted in Table 6. For example, in the basic experiment, all statistically significant effects were covered by the model, indicating to the solar-cell developer which parameters have the greatest influence on output power.

Table 6. Effect Summary Model Fit for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Source	LogWorth	PValue
basethick	241.699	0.00000
botbasethick	155.315	0.00000
botbasedopconc	151.940	0.00000
winthick	151.861	0.00000
basethick*basethick	114.921	0.00000
botbasethick*botbasethick	95.407	0.00000
winthick*basethick	84.372	0.00000
botemdopconc	59.962	0.00000
buffdopconc	54.548	0.00000
basethick*botbasedopconc	38.541	0.00000
basethick*botbasethick	37.066	0.00000
emdopconc	36.761	0.00000
emdopconc*basethick	33.616	0.00000
botbasethick*botbasedopconc	30.398	0.00000
botwinthick	14.640	0.00000
windopconc	14.054	0.00000
windopconc*basethick	12.888	0.00000
botemthick*botemdopconc	12.623	0.00000
winthick*emdopconc	12.104	0.00000
buffdopconc*buffdopconc	11.473	0.00000
botemthick	6.293	0.00000 ^
basethick*buffthick	5.767	0.00000
emthick*basethick	5.302	0.00000
winthick*botbasethick	4.855	0.00001
emthick	4.135	0.00007 ^
basethick*bsfthick	3.935	0.00012
basethick*botemdopconc	3.594	0.00025
windopconc*windopconc	3.050	0.00089
botbasedopconc*botbasedopconc	2.480	0.00331
bsfthick	1.092	0.08088 ^
buffthick	0.548	0.28337 ^

Compared to the basic experiment, this model had more parameters and interactions with a statistically significant effect on output power. This was expected, as more samples increase statistical power.

In Table 7, the RSquare value is slightly higher than the one in the basic experiment. This has to do with the greater number of parameters and their interactions in the model. This model explains 84.5% of the variance.

Table 7. Summary of Fit Model for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Summary of Fit	
RSquare	0.845391
RSquare Adj	0.842009
Root Mean Square Error	0.905651
Mean of Response	33.61957
Observations (or Sum Wgts)	1449

With the prediction profiler, a kind of sensitivity analysis was conducted to learn more about the influence of each factor on the overall result. The winthick parameter is a good example of the rationale behind this process. As depicted in Figure 24, the best output values are achieved if the winthick parameter is at the lowest end of the experimentation input value range. In the next iteration of the development process for that specific solar cell, the value range for winthick should be lowered to capture the best range of values for that factor.

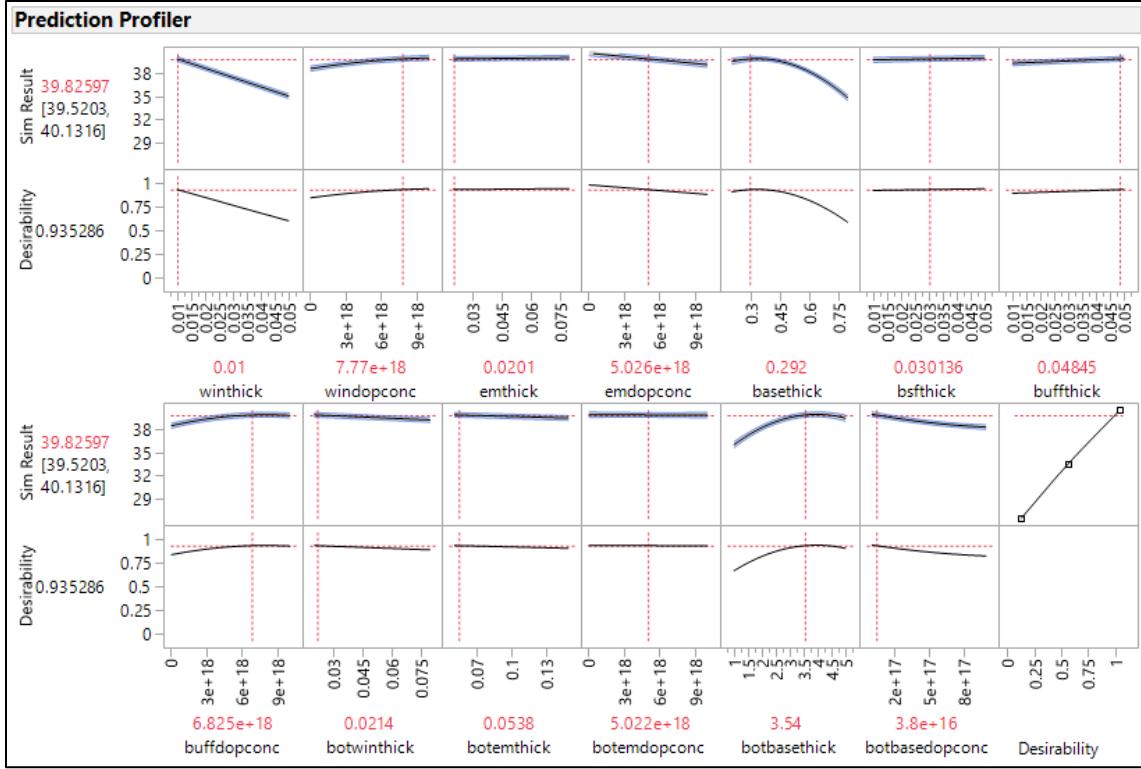


Figure 24. Prediction Profiler for Response Surface Improved 1,542-Design-Point Agui Cell Design Optimization for Linear-Transformed Dopants.

c. Conclusion

With the analysis of the rotated and stacked design, it appears that the technique provides much better coverage of the experimentation space and detects trends in the data better. Therefore, the predicting capability for the fitted model is better. Of course, it is up to the developer to decide what is good enough to support his work. With rotating and stacking, there is a scalable method to adjust for requirements of the solar cell's current development phase.

3. Improved Agui Solar Cell with 1,542 Design Points (Rotated and Stacked)

Next, the improved solar cell marks the second step in an iterative process to improve the design of the cell further. For each factor, the ranges have been adapted to produce a much denser coverage around the areas that deliver the

best output values. In the case of the Agui cell design, input values from the row that produced the highest output value were the basis for creating a narrower wrapper than the one in the previous experiment. The result of this technique is discussed in the following paragraphs.

a. General Data Screening and Analysis

Comparing the design point distribution from Figure 22 with that of Figure 25, we see differences in how the values are dispersed in the space. For example, the *winthick* value was more or less a continuous value in the previous experiment. In the case of this experiment, the developers chose to limit the value space to a few discrete values. However, the developer has to consider many options during this process. Due to the complexity of the subject, this section does not discuss those considerations at length. The best advice for developers is to familiarize themselves with the possibilities of this new approach and use all relevant information as appropriate.

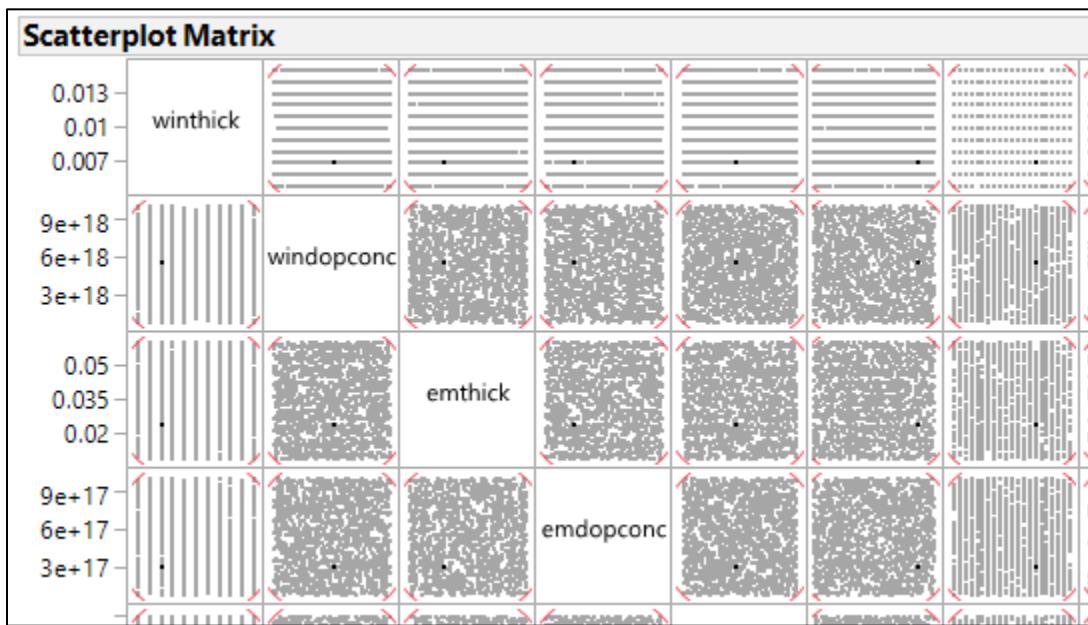


Figure 25. Extract of Scatterplot Matrix for Improved 1,542-Design-Point Agui Cell Design Using Linear Transformed Dopants.

The distribution of the output power values in Figure 26 gives a completely different picture than that depicted in Figure 23. The shape deviates from the normal distributed type of curve in Figure 23 to a curve concentrated toward the right side. This distribution indicates a new stage in which the developer shifted the output power distribution in the desired direction. The mean of 38.4 mW/cm², compared to the previously recorded 33.6 mW/cm², is a clear indication. In addition, nearly all simulation runs converged. A convergence of 1532 out of 1542 runs is a very good outcome, indicating that most of the input factors were already in the active range of the design.

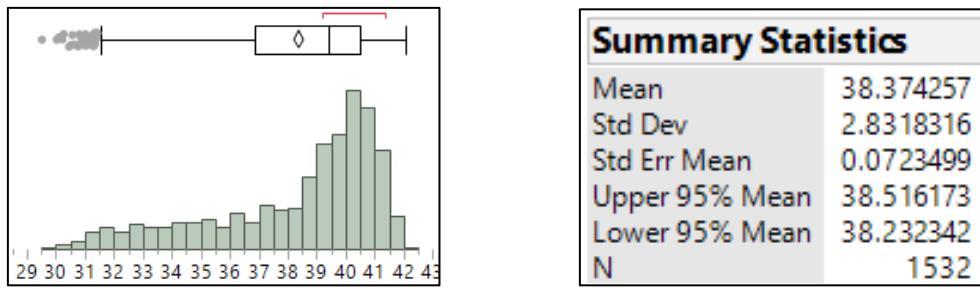


Figure 26. Simulation Output Power Distribution for Improved 1,542-Design-Point Agui Cell Design Using Linear Transformed Dopants.

b. Model Fit and Response Surface Analysis

The effect summary of the experiment in Table 8 shows a decrease of the LogWorth values compared to those in Table 6. Since the stepwise regression model fit selects the input factor values that have the greatest effect on the output power, lower LogWorth values indicate only slightly less influence on the simulation output value than in the previous model. As anticipated, the smaller region is dominated by a smaller number of factors.

Table 8. Effect Summary Model Fit Improvements for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Source	LogWorth	PValue
basethick	37.648	0.00000
tunemthick*tunemthick	14.526	0.00000
tunemthick	7.432	0.00000 ^
basethick*tunemthick	4.730	0.00002
botwindopconc	3.451	0.00035
tunemthick*botbasedopconc	3.345	0.00045
buffthick	2.307	0.00493
botwindopconc*botwindopconc	2.122	0.00754
botbasedopconc	1.920	0.01203 ^
basethick*botbuffthick	1.614	0.02432
basethick*botwindopconc	1.586	0.02592
buffthick*tunemthick	1.486	0.03263
basethick*botbsfthick	1.443	0.03606
botbasedopconc*botbasedopconc	1.405	0.03936
basedopconc*botemdopconc	1.345	0.04519
buffdopconc*botbasedopconc	1.323	0.04750
buffthick*buffthick	1.315	0.04841
basethick*botbasedopconc	1.302	0.04990
buffthick*botbasethick	1.229	0.05900
botbuffthick*botbuffthick	1.165	0.06839
botbsfthick*botbsfthick	1.153	0.07028
tunemdopconc	1.089	0.08152
botbasethick*botbasethick	1.074	0.08440
tunemthick*botwindopconc	0.980	0.10483
emthick	0.946	0.11316
buffthick*botwindopconc	0.893	0.12782
emthick*buffthick	0.862	0.13742
botemdopconc*botbuffthick	0.691	0.20389
basedopconc	0.653	0.22237 ^
buffdopconc	0.326	0.47191 ^
botbuffthick	0.299	0.50287 ^
botbasethick	0.216	0.60823 ^
botemdopconc	0.202	0.62865 ^
botbsfthick	0.138	0.72769 ^

The summary of fit in Table 9 shows an RSquare of only 0.22. The low value cannot be explained, so it requires further analysis.

Table 9. Summary of Fit Model Improvements for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Summary of Fit	
RSquare	0.216948
RSquare Adj	0.199163
Root Mean Square Error	2.534191
Mean of Response	38.37426
Observations (or Sum Wgts)	1532

To explain the low RSquare value a residual plot might be helpful to identify the reason for the bad model fit. The residual plot in Figure 27 does not reveal any irregularities and does not explain the low RSquare value.

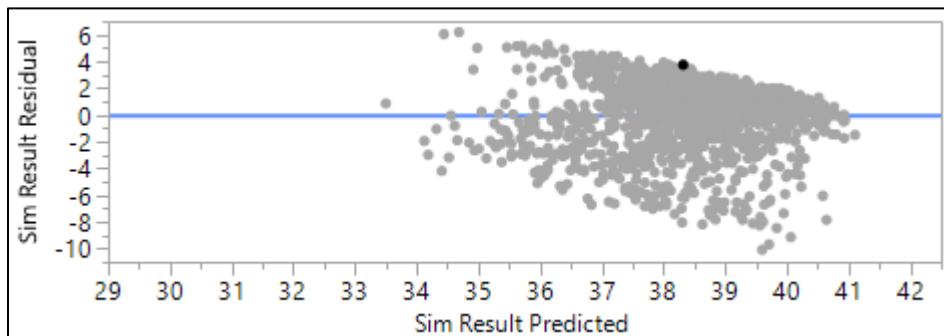


Figure 27. Residual by Prediction Plot for 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

The prediction profiler, depicted in Figure 28, identifies the parameters that have the greatest influence on the output as well as the direction in which they need altering to produce the desired effect. In the case of that experiment, the basethick and tunemthick values had the greatest effect on the output power value and required exploration in the next iteration to increase the odds of producing a higher output value. The predicted value of the model increased from 39.83 mW/cm² with a confidence interval of 39.53 mW/cm² to 40.13 mW/cm² in Figure 24 to 43.16 mW/cm² with a confidence interval of 42.25 mW/cm² to 44.06 mW/cm² in Figure 28, indicating how the regression model detected the shape of the response surface.

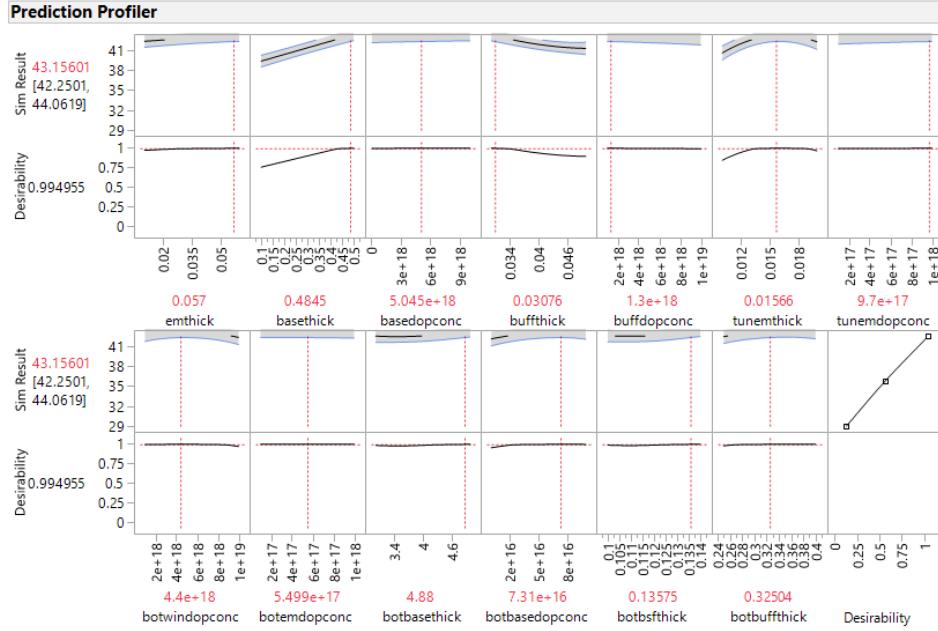


Figure 28. Prediction Profiler for Response Surface Improved 1,542-Design-Point Agui Cell Design Optimization Using Linear-Transformed Dopants.

c. Conclusion

The second step of this new approach confirmed the expected behavior of the condensed design space with narrower factor ranges. A higher output power value was predicted, and the general trend of the data, depicted in Figure 26, shifted to the right. This was the expected outcome of bounding the response surface to the region of highest output. The regression model gives for each statistically significant input factor the direction for the parameter's range in the simulation's next iteration. The reduced number of parameters in the model indicated that only a few affected the output power value.

4. Second Iteration of Improved Agui Solar Cell with 1,542 Design Points (Rotated and Stacked)

The second iteration of the improved solar cell illustrated some of the features the new process included by design. For each single input value, the ranges were adapted again for much denser coverage around the areas that

produced the best output values, and the granularity of some of the parameters was increased as well. The procedure was the same as described in the previous section.

a. General Data Screening and Analysis

Comparing the design-point distribution from Figure 25 and the one in Figure 29 shows differences in how the values were dispersed in the space. For example, the winthick values in the first column exhibited some discrete values in the previous experiment but reverted to continuous numbers in the current experiment.

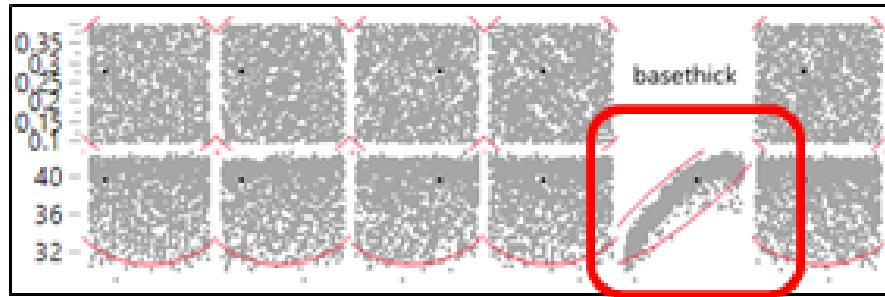


Figure 29. Extract of Scatterplot Matrix from Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Another interesting aspect of the scatterplot is the distribution of the simulation results in the basethick column. As shown, the simulation result increases with increases of the basethick input value, which is precisely the information a developer needs to focus the new design. Still, this step in the data analysis is meant primarily to scrutinize the data to see whether the structure is as intended.

The distribution of the output power values does not change much from Figure 26 to Figure 30. The shape still tends toward the right side. The mean of 38.27 mW/cm^2 with a confidence interval of 38.11 mW/cm^2 to 38.42 mW/cm^2 is

0.1 mW/cm² less than in the previous experiment, but the maximum output power value has improved to 42.86 mW/cm². Even more simulation runs, 1,536 out of 1542, converged, indicating that most of the input factors were already in the active range of the design. Developers should also consider exploring the outlier output values to learn what might have caused the values so far from the mean.

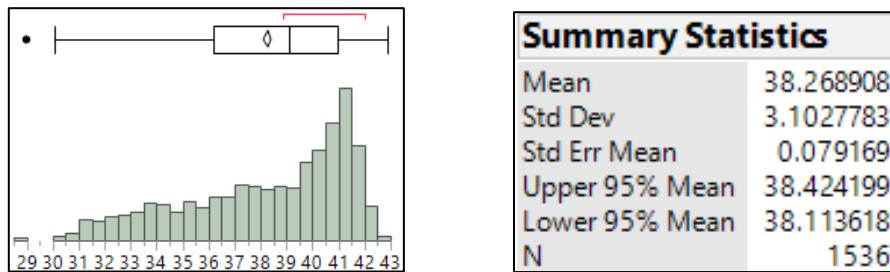


Figure 30. Simulation Output Power Distribution for Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

b. Model Fit and Response Surface Analysis

The effect summary in Table 10 shows more or less the same picture as in the previous experiments. The expectation was that this run would lower LogWorth values of factors in the model again. However, this iteration of the simulation did not improve the fitted model. Although a new maximum output power value was produced, there is still the potential for obtaining a higher value.

Table 10. Effect Summary Model Fit for Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	27.302997	0.152415	179.14	<.0001*
windopconc	5.384e-20	4.45e-21	12.11	<.0001*
emthick	36.650918	2.004002	18.29	<.0001*
emdopconc	1.367e-19	4.45e-20	3.07	0.0022*
basethick	32.127208	0.133525	240.61	<.0001*
basedopconc	-7.26e-20	6.67e-21	-10.89	<.0001*
bsfthick	29.135152	1.289787	22.59	<.0001*
bsfdopconc	-1.23e-19	4.45e-20	-2.78	0.0056*
buffthick	30.863259	1.335335	23.11	<.0001*
buffdopconc	9.271e-20	4.45e-21	20.85	<.0001*
tunbasethick	11.82139	1.333312	8.87	<.0001*
botbasethick	0.0226297	0.013362	1.69	0.0905
botbasedopconc	-1.42e-18	2e-19	-7.11	<.0001*
botbsfthick	1.2048242	0.444574	2.71	0.0068*
botbsfdopconc	2.498e-18	9.53e-20	26.21	<.0001*
botbuffdopconc	3.574e-21	9.31e-22	3.84	0.0001*
(emthick-0.02499)*(basethick-0.24983)	-316.5303	23.9631	-13.21	<.0001*
(emthick-0.02499)*(buffthick-0.04002)	-1072.584	220.2662	-4.87	<.0001*
(basethick-0.24983)*(bsfthick-0.02949)	-270.0123	15.80987	-17.08	<.0001*
(basethick-0.24983)*(buffthick-0.04002)	-259.708	15.79922	-16.44	<.0001*
(basethick-0.24983)*(buffdopconc-5.5e+18)	2.999e-19	5.07e-20	5.91	<.0001*
(basethick-0.24983)*(tunbasethick-0.02502)	-103.9238	16.25765	-6.39	<.0001*
(basethick-0.24983)*(botbasethick-4.50093)	1.3167127	0.161752	8.14	<.0001*
(basethick-0.24983)*(botbasedopconc-1e+17)	-1.24e-17	2.16e-18	-5.73	<.0001*
(basethick-0.24983)*(botbsfdopconc-2.9e+17)	7.292e-18	1.06e-18	6.89	<.0001*
(basedopconc-7e+18)*(buffthick-0.04002)	2.135e-18	7.46e-19	2.86	0.0043*
(basedopconc-7e+18)*(buffdopconc-5.5e+18)	1.481e-38	2.63e-39	5.63	<.0001*
(bsfthick-0.02949)*(buffthick-0.04002)	-578.4871	154.9027	-3.73	0.0002*
(bsfdopconc-5.5e+17)*(botbsfdopconc-2.9e+17)	1.241e-36	3.8e-37	3.27	0.0011*
(buffthick-0.04002)*(buffdopconc-5.5e+18)	-4.58e-18	5.48e-19	-8.36	<.0001*
(botbasedopconc-1e+17)*(botbsfdopconc-2.9e+17)	-6.36e-36	1.64e-36	-3.88	0.0001*
(botbsfthick-0.12507)*(botbsfdopconc-2.9e+17)	-3.21e-17	3.83e-18	-8.38	<.0001*
(botbsfdopconc-2.9e+17)*(botbuffdopconc-2.8e+19)	-2.64e-38	7.26e-39	-3.64	0.0003*
(windopconc-5.5e+18)*(windopconc-5.5e+18)	-8.27e-39	1.93e-39	-4.29	<.0001*
(basethick-0.24983)*(basethick-0.24983)	-140.6229	1.747618	-80.47	<.0001*
(buffdopconc-5.5e+18)*(buffdopconc-5.5e+18)	-2.48e-38	1.93e-39	-12.87	<.0001*
(botbsfthick-0.12507)*(botbsfthick-0.12507)	-56.36965	19.32955	-2.92	0.0036*
(botbsfdopconc-2.9e+17)*(botbsfdopconc-2.9e+17)	-1.9e-35	8.9e-37	-21.36	<.0001*

The summary of fit in Table 11 gives a very high value in this iteration. Compared to the value obtained in the previous experiment, this is better and an indication the model is more reliable in its prediction power.

Table 11. Summary of Fit Model for Second Iteration of Improved 1,542-Design-Point Agui Cell Design Using Linear-Transformed Dopants.

Summary of Fit	
RSquare	0.979058
RSquare Adj	0.97854
Root Mean Square Error	0.454529
Mean of Response	38.26891
Observations (or Sum Wgts)	1536

The prediction profiler, as shown in Figure 31, identifies the parameters that have the greatest influence on the output and the direction in which they have to be altered to produce the desired effect. In the case of this experiment, the basethick values still have the greatest influence. Considering the prediction value of 43.11 mW/cm^2 , the metamodel does detect the response surface shape of the simulated solar-cell design much better than the previous model.

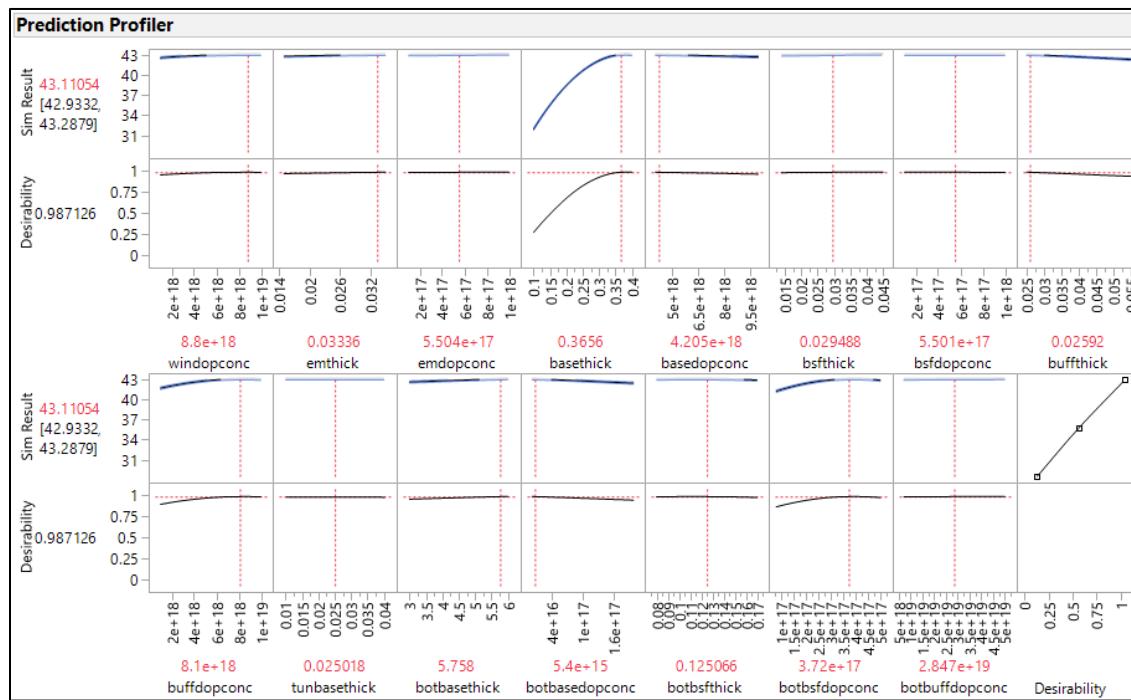


Figure 31. Prediction Profiler for Response Surface in Second Iteration of Improved 1,542-Design-Point Agui Cell Design Optimization Using Linear-Transformed Dopants.

c. Conclusion

In the third step of this new approach, the expected behavior of the condensed design space with narrower parameter ranges proved to work. A new maximum output power value was generated by the simulation system, and the general trend of the output data, depicted in Figure 30, still shifted to the right toward higher output power values. This was the intended outcome of bounding the response surface in an area of the highest output value. The regression model gives for each statistically significant input factor the direction for determining the range of that parameter for the next iteration. The prediction profiler and the height of the LogWorth values suggest the benefit of continuing the optimization process to obtain better values.

Figure 32 gives the I–V curve derived from the simulation runs that produced the highest output power value and shows the improvement of the cathode current flow, from the AM1.5 genetic algorithm–optimized design to the design optimized with the NOLH DoE. Solar-cell developers are particularly interested in deriving information from the shape of I–V curves.

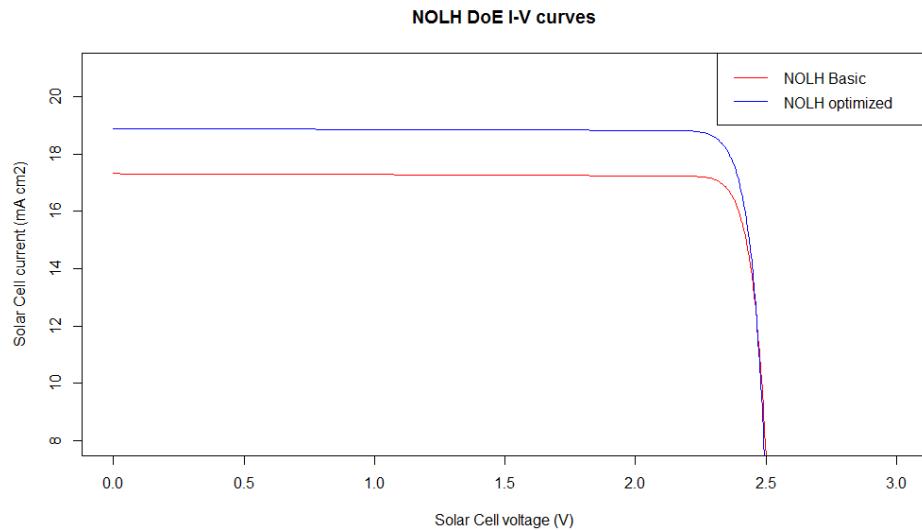


Figure 32. Current vs. Voltage (I–V Curves) for the NOLH Basic and Optimized Two-Junction Solar-Cell Designs Generated from the Maximum Output Power Simulation Run.

V. CONCLUSION AND RECOMMENDATION

This thesis introduced a new approach using the NOLH DoE to optimize multi-junction solar cells and demonstrated its success. Many of the limitations that come with the use of the genetic algorithm were eliminated. The major shortcomings of the genetic algorithm include the complexity in setting up all the parameters of the design, the inflexibility in addressing cell design changes, and the inefficient use of the simulation data to analyze the input/output relationship of the model. The latter is the only way to see whether the simulation is doing exactly what it is designed to do. Another limitation is that the genetic algorithm allows only sequential simulation runs, so more complex studies require longer runtimes.

The complexity issue of the genetic algorithm was mitigated using the easy-to-implement NOLH Excel spreadsheet provided by the NPS SEED center. As long as the spreadsheet design permits, further simulation parameters can be added and/or their ranges changed. In addition, the granularity was increased using the NOLH design—highlighted by the 257 design points of the NOLH versus the 128 intermediate variable parameters of the genetic algorithm introduced by Tsutagawa (2013). Using the rotating and stacking technique achieves much better space filling.

The analysis of the input/output data relationship unveils new possibilities for understanding which factors affected the response, and how. It was the first time this powerful tool was introduced in the solar-cell development process, and it guided the changes of the design effectively. Using JMP statistical analysis software makes exploring the data easy, ultimately leading to a tremendous decrease in design turnaround time. No separate experimentation was necessary to determine the course of the next design step. In addition, determining the initial parameter ranges became straightforward. It is recommended that researchers use a wide experimentation space at the beginning and narrow the intervals in subsequent iterations. In this regard, the NOLH DoE demonstrates

the advantage of revealing parameter interactions. If in the utilization of the genetic algorithm a mechanism is utilized to track the input and output data, a basic data analysis may be possible.

The design's pre-generating property in the new approach solved the runtime issue. Although the NPS high performance computer Hamming was not available, this thesis established its theoretical applicability. Processing different simulation designs in parallel, limited at NPS by the number of available Silvaco Atlas software licenses, could have reduced the runtime by at least an order of magnitude. Moreover, by using Hamming, additional processors may have been available to reduce the runtime of each single simulation run, since Silvaco Atlas enables symmetric multiprocessing. A runtime reduction from several hours or even days to only a few hours or minutes drastically accelerates cell development and turnaround times. The potential of this reduction can only be adumbrated. The Hamming implementation should be considered in future work.

The most important achievement of such a process is the realization of a desired simulation result, which in this case was the increase of power output from a dual-junction solar cell. The basic design optimized the genetic algorithm for an AM1.5 spectrum, taken from the work of Tsutagawa (2013) in his dissertation and rebuilt by R. Kilway, personal communication, March 24, 2017, yielding an output value of 35.76 mW/cm^2 . The design was improved to 39.58 mW/cm^2 in the basis application run and improved again by rotating and stacking to 40.38 mW/cm^2 . In the first run with the condensed experimentation space, a value of 42.1 mW/cm^2 was attained. That is a major achievement since it compares the value from the initial run, which is already an AM0-optimized design, with the second iteration in this process. The second iteration to improve the design obtained an output power value of 42.86 mW/cm^2 , which is again an increase, and the data analysis indicates potential for higher values. The first re-run of the experiment accomplished a noticeable increase in the power performance of the solar-cell design.

Due to the easy overall process and the simulation results, the NOLH DoE should become the new standard for optimizing solar cells with Silvaco Atlas; no major obstacles were identified that require future work with the exception of getting Hamming up and running.

Another wide area of further work is in the area of physical implementation of the optimized solar-cell designs and the performance in comparing the simulated models. Considering manufacturing tolerances, creating a consumer product ready with increased efficiency has a long way to go.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. SILVACO ATLAS TEMPLATE FILE

```
#Recreating Michal. multijunction cell, pg 122 (InGaP/GaAs dual MJ cell)
```

```
go atlas simflags="-P 6"

set winthick= 0.03
set windopconc= 2E18
set emthick= 0.05
set emdopconc= 2E18
set basethick= 0.55
set basedopconc= 1.5E17
set bsfthick= 0.03
set bsfdopconc= 2E18
set buffthick= 0.03
set buffdopconc= 1E18
set tunemthick= 0.015
set tunemdopconc= 8E18
set tunbasethick= 0.015
set tunbasedopconc= 1E19
set botwinthick= 0.05
set botwindopconc= 1E19
set botemthick= 0.1
set botemdopconc= 2E18
set botbasethick= 3
set botbasedopconc= 1E17
set botbsfthick= 0.1
set botbsfdopconc= 2E18
set botbuffthick= 0.3
set botbuffdopconc= 7E18

mesh auto
x.m loc=0.0 s=0.25
x.m loc=1.0 s=0.25

#top cell
region name="n+ type AlInP" material=AlInP bot thick=$winthick donors=$windopconc ny=10
x.comp=0.52
region name="n+ type InGaP" material=InGaP bot thick=$emthick donors=$emdopconc ny=10
x.comp=0.51
region name="p+ type InGaP" material=InGaP bot thick=$basethick accept=$basedopconc
ny=10 x.comp=0.51
region name="p+ type InGaP" material=InGaP bot thick=$bsfthick accept=$bsfdopconc ny=10

region name="p+ type AlInP" material=AlInP bot thick=$buffthick accept=$buffdopconc ny=10
x.comp=0.52

#tunnel

region name="p+ type InGaP" material=InGaP bot thick=$tunemthick accept=$tunemdopconc
ny=100 x.comp=0.51
region name="n+ type InGaP" material=InGaP bot thick=$tunbasethick
donors=$tunbasedopconc ny=100 x.comp=0.51
```

```

#bottom cell

region name="n+ type AlInP" material=AlInP      bot thick=$botwinthick donors=$botwindopconc
ny=10 x.comp=0.52
region name="n+ type GaAs" material=GaAs        bot thick=$botemthick donors=$botemdopconc
ny=10
region name="p+ type GaAs" material=GaAs        bot thick=$botbasethick
acceptors=$botbasedopconc ny=10
region name="p+ type InGaP" material=InGaP    bot thick=$botbsfthick acceptors=$botbsfdopconc
ny=10 x.comp=0.51
region name="p+ type GaAs" material=GaAs    bot thick=$botbuffthick acceptors=$botbuffdopconc
ny=10
electrode name=cathode material=AlInP top
electrode name=tunnel material=InGaP
y.min=$winthick+$emthick+$basethick+$bsfthick+$buffthick
y.max=$winthick+$emthick+$basethick+$bsfthick+$buffthick+$unemthick+$unbasesthick
electrode name=anode material=GaAs bottom
contact name=tunnel resist=1E17

material mat=GaAs sopra=Gaas.nk
material material=InGaP sopra=Againp0_mod.nk
material material=Ge sopra=Ge.nk
material material=AlInP sopra=Againp10_mod.nk

#quit

material material=InGaP EG300=1.9 PERMITTIVITY=11.8 AFFINITY=4.09
material material=InGaP MUN=3500 MUP=400
material material=InGaP NC300=9.3268e17 NV300=9.3268e18 NI=323
material material=InGaP AUGN=3e-30 AUGP=3e-30 COPT=1e-10 TAUN=1e-03 TAUP=1e-03
material material=Vacuum real.index=3.3 imag.index=0
material material=AlInP EG300=2.4 PERMITTIVITY=12.5 AFFINITY=4.04
material material=AlInP MUN=3000 MUP=150
material material=AlInP NC300=6.617e20 NV300=6.617e21 NI=100
material material=AlInP AUGN=5.447e-30 AUGP=2.957e-29 COPT=1e-10 TAUN=1e-03
TAUP=1e-03
material material=GaAs EG300=1.424 PERMITTIVITY=12.9 AFFINITY=4.07
material material=GaAs MUN=8500 MUP=400
material material=GaAs NC300=4.7e17 NV300=9e18
material material=GaAs AUGN=1e-30 AUGP=1e-30 COPT=7.2e-10 TAUN=1e-03 TAUP=1e-03
models srh auger optr fermi connmob bgn temp=300 print
method itlimit=40 maxtraps=20
beam num=1 x.origin=0 y.origin=-5 angle=90 back.refl wavel.start=0.12 wavel.end=3.455
wavel.num=1500 AM0

solve init
solve b1=1e-01
solve b1=1
save outf=Agui_Dual_Resist.str

log outf=Agui_Dual_Resist.log
save outf=0V.str
solve previous
solve vanode=0.001

```

```
solve vanode=0.05
solve vstep=0.05 vfinal=2.1 name=anode
solve vanode=2.1 vstep=0.005 vfinal=2.7 name=anode

extract init infile="Agui_Dual_Resist.log"
extract name="Jsc" max(curve(v."anode", i."cathode"))
extract name="Jsc_mAcm2" $Jsc*1e08*1e03
extract name="Voc" x.val from curve(v."anode", i."cathode") where y.val=0.0
extract name="Pm" max(curve(v."anode", (v."anode" * i."cathode")))
extract name="Pmax_mW/cm2" $Pm*1e08*1e03
extract name="Vm" x.val from curve(v."anode", (v."anode"*i."cathode")) where y.val=$"Pm"
extract name="Im_mAcm2" $Pm/$Vm*1e08*1e03
extract name="FF" ($Pm/($Jsc*$"Voc"))*100
extract name="intens" max(beam."1")
extract name="Eff" (1e8*$Pm/$intens)*100
extract name="iv" curve(v."anode", i."cathode"*1e08*1e03) outfile="Agui_Dual_Resist.dat"

quit
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. PYTHON SCRIPTS

A. PYTHON SCRIPT FOR SILVACO ATLAS FILE PRE-PROCESSING

```
# -*- coding: utf-8 -*-

Created on Wed Mar 22 21:41:53 2017

@author: Silvio

"""

import codecs      # Keeping the Silvaco file in the right format
import csv         # Read in the CSV

with open("Sim Input lin transf and stacked secound iteration.csv", "r") as csvfile:

    Input_data = csv.reader(csvfile)           # Save the CSV in a variable

    for row_index, row in enumerate (Input_data): # Looping the Input data matrix by row and
                                                # generating the row index

        with codecs.open("Japanize Solar Silvaco Template.in",'r',encoding='utf8') as f: # Open the
            default Silvaco scriped

            lines = f.readlines()   # Read the scriped line by line

            for index in range (3, 3 +len(row)):  # index the line range where to change the parameter

                parameter, value = lines[index].split("=", 1) # split the line at the "=" chr

                if (index & 1) == 1: # Selecting the doping level lines in the skript

                    lines[index] = parameter + "=" + " " + str(row[index - 3]) + "\n" # Build the new content

                else:

                    lines[index] = parameter + "=" + " " + str(float(1E16) * float(row[index - 3])) + "\n" # Linear
                        # transformation of the doping levels

            for out in [88, 90, 99, 110]:          # Exception handling

                if out == 110:

                    resitual = lines[out].split(".") # split the line at the . chrs

                    lines[out] = resitual[0] + "." + resitual[1] + "." + resitual[2] + str(row_index + 1) + "." +
                    resitual[3].rstrip("\n") + "\n" # Build the new content

                else:

                    resitual = lines[out].split(".") # split the line at the . chr
```

```
    lines[out] = resitual[0] + str(row_index + 1) + "." + resitual[1].rstrip("\n") + "\n" # Build the
new content

    with
codecs.open("Silvaco_Japanize_Solar_Design_Input_linear_transformed_and_stacked_second_i
teration" + str(row_index + 1) + ".in",'w',encoding='utf8') as file: # Open a new file for each design

    for line in lines:

        file.write(line) # Write the lines into the file

    file.close()
```

B. PYTHON SCRIPT TO RUN THE SILVACO ATLAS FILE

```
# -*- coding: utf-8 -*-

"""
Created on Thu Apr 6 19:44:11 2017

@author: Silvio

"""

import subprocess          # Required to call other programs
import time                # Required to get the system time

timetracker = []           # Time capturing vector

for filenumber in range(1,1543,1):      # Run of the number of designs

    outputcmd = "deckbuild -run"
    Silvaco_Japanize_Solar_Design_Input_linear_transformed_and_stacked" + str(filenumber) + ".in"
        # Specifying the design to run in Silvaco

    start = time.time()          # Capturing the start time
    subprocess.check_output(outputcmd,shell=True)      # Call Silvaco to run the design
    timetracker.append(time.time() - start)      # Appending the runtime to the time vector
```

C. PYTHON SCRIPT FOR SILVACO ATLAS OUTPUT VALUE POST-PROCESSING

```
# -*- coding: utf-8 -*-

"""
Created on Mon Apr 10 20:13:52 2017

@author: Silvio

"""

import numpy as np          # Required to apply multiplication to complete vector
import pandas as pd         # Required to read in CSV file

maxpower = []               # Vector to preserve the maximum output power

for filenumber in range(1, 1543, 1):

    arr = np.loadtxt('Agui_Dual_Resist' + str(filenumber) + '.dat', delimiter=' ', skiprows = 4)
    # Read the output value file from Silvaco

    maxpower.append (max(np.multiply(arr[:, 0], arr[:, 1]))) # Compute the max power output value

df = pd.read_csv('Japanize Solar Design Input linear transformed and stacked.csv',
header=None)           # Read the input value CSV file

for c in df.columns:

    if (c & 1) == 1:

        df[c] = df[c].apply( lambda x: (float(1E16) * x)) # Linear transformation of doping levels

new_column = pd.DataFrame(maxpower) # Change vector into pandas data frame column

df = df.merge(new_column, left_index = True, right_index = True) # Merge input and output data
df.to_csv('AAASimulation result linear transformation and stacked.csv',header=False,
index=False)           # Write the combined data back to a CSV ready for analysis
```

LIST OF REFERENCES

Agui, T., Takamoto, T., Ikeda, E., & Kurita, H. (1998). *High-efficient dual-junction InGaP/GaAs solar cells with improved tunnel interconnect*. Proceedings of the International Conference on Indium Phosphide and Related Materials, 203–206. doi: 10.1109/ICIPRM.1998.712437

Ang, J. K. (2006). *Extending orthogonal and nearly orthogonal Latin hypercube designs for computer simulation and experimentation*. Master's thesis, Naval Postgraduate School, Monterey, CA.

Bates, A. D. (2004a). *Novel optimization techniques for multijunction solar cell design using Silvaco Atlas*. Master's thesis, Naval Postgraduate School, Monterey, CA.

Bates, A. D. (2004b). *Novel optimization techniques for multijunction solar cell design using Silvaco Atlas*. Master's thesis, Naval Postgraduate School, Monterey, CA. p 10-11

Box, G. E. P., & Behnken, D. W. (1960). Some new three level designs for the study of quantitative variables. *Technometrics*, 2, 455–475.

Box, G. E. P., & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journals of the Royal Statistical Society, Series B: Statistical Methodology*, 13, 1–45.

Celani (2013). *LT3652 Solar Battery Charger Maintains High Efficiency in Low Light*. Linear Technology Corporation. Milpitas, CA

Cioppa, T. M., & Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics*, 49(1), 45–55.
doi:10.1198/004017006000000453

Darwin, C., & Wallace, A. (1858). On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. *Journal of the Proceedings of the Linnean Society of London, Zoology* 3 (pp. 45–62). http://darwin-online.org.uk/converted/pdf/1858_species_F350.pdf

Depletion region. (2015). Retrieved May 23, 2017, from www.physics-and-radio-electronics.com/electronic-devices-and-circuits/semiconductor-diodes/depletion-region.html

Doping: N- and p-semiconductors. (n.d.). Retrieved May 23, 2017, from www.halbleiter.org/en/fundamentals/doping/

Eurems Europe NV. (n.d.). Technical Info PV Cells. Retrieved from www.eurems.com/links-topics/technical-info/cells

Fang, K. T. (1980). The uniform design: Application of number-theoretic methods in experimental design. *Acta Mathematicae Applicatae Sinica*, 3, 363–372.

Fisher, R. A. (1925). *Statistical methods for research workers*. Biological Monographs and Manuals Series. Edinburgh, Scotland: Oliver and Boyd.

Florian, A. (1992). An efficient sampling scheme: Updated Latin hypercube sampling. *Probabilistic Engineering Mechanics*, 7, 123–130.

Genetic algorithms tutorial. (2017). Retrieved May 23, 2017, from www.tutorialspoint.com/genetic_algorithms

Green, M. (2002). *The verification of Silvaco as a solar cell simulation tool and the design and optimization of a four-junction solar cell*. Master's thesis, Naval Postgraduate School, Monterey, CA.

Green, M. A. (2003). *Third generation photovoltaics: Advanced solar energy conversion*. Berlin: Springer.

Hernandez, A. S. (2008). *Breaking barriers to design dimensions in nearly orthogonal Latin hypercubes*. Doctoral dissertation, Naval Postgraduate School, Monterey, CA.

Hernandez, A. S., Lucas, T. W., & Carlyle, M. (2012). Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination. *ACM Transactions on Modeling and Computer Simulation*, 22(4), 20:1–17.

Hoke, A. T. (1974). Economical second-order designs based on irregular fractions of the 3ⁿ factorial. *Technometrics*, 16(3), 375–384.

Hu, C. (2009). *Electrons and Holes in Semiconductors*. P 9. Retrieved May 28, 2017, from https://people.eecs.berkeley.edu/~hu/Chenming-Hu_ch1.pdf

Iman, R., & Conover, W. J. (1982). A distribution-free approach to including rank correlation among input variables. *Communication in Statistics – Simulation Computations*, 11(3), 311–334.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maxmin distance design. *Journal of Statistical Planning and Inference*, 26, 131–148.

Kiefer, J., & Wolfowitz, J. (1959). Optimal designs in regression problems. *Annals of Mathematical Statistics*, 30, 271–294.

MacCalman, A. D. (2013). *Flexible space-filling designs for complex system simulations*. Doctoral dissertation, Naval Postgraduate School, Monterey, CA.

MacCalman, A. D., Vieira, H., & Lucas, T. (2017). Second-order nearly orthogonal Latin hypercubes for exploring stochastic simulations. *Journal of Simulation*, 11(2), 137–150. doi: 10.1057/jos.2016.8

The MathWorks Inc. (n.d.). MATLAB: analysis and design computer language [Computer software], Available at <https://www.mathworks.com/products/matlab.html>

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from computer code. *Technometrics*, 42(1), 239–245.

Michalopoulos, P. (2002). *A novel approach for the development and optimization of state-of-the-art photovoltaic devices using Silvaco*. Master's thesis, Naval Postgraduate School, Monterey, CA.

National Renewable Energy Laboratory. (2017). Retrieved May 28, 2017, from <https://www.nrel.gov/pv/assets/images/efficiency-chart.png>

Oracle INC. (n.d.). Java: general-purpose computer programming language. [Computer software], Available at <https://www.java.com/en/>

Owen, A. B. (1994). Controlling correlations in Latin hypercube samples. *Journal of the American Statistical Association: Theory and Methods*, 89(428), 1517–1522.

Pauli exclusion principle. (2016). In Encyclopædia Britannica. Retrieved May 23, 2017, from www.britannica.com/science/Pauli-exclusion-principle

Periodic Table of Elements. (2014). Retrieved June 14, 2017, from <http://scienzenotes.org/wp-content/uploads/2014/05/SciNotesPeriodicTable.png>

Physics and Radio-Electronics. (2014a). *Atomic structure of intrinsic semiconductor materials Silicon and Germanium*. Retrieved May 23, 2017, from www.physics-and-radio-electronics.com/electronic-devices-and-circuits/semiconductor/intrinsic-semiconductor/atomic-structure-of-silicon-and-germanium.html

Physics and Radio-Electronics. (2014b). *Electron and hole current*. Retrieved June 14, 2017, from www.physics-and-radio-electronics.com/electronic-devices-and-circuits/semiconductor/intrinsic-semiconductor/atomic-structure-of-silicon-and-germanium.html

Python Software Foundation. (n.d.). Python: general-purpose computer programming language. [Computer software], Available at <https://www.python.org/downloads/>

Roquemore, K. G. (1976). Hybrid designs for quadratic response surfaces. *Technometrics*, 18(4), 419–423.

Sanchez, S. M., & Sanchez, P. J. (2005). Very large fractional factorial and central composite designs. *ACM Transactions on Modeling and Computer Simulation*, 15(4), 362–377.

Sanchez, S. M. (2011). NOLH designs [Excel spreadsheet]. Retrieved February 25, 2017, from <http://my.nps.edu/web/seed/software-downloads>

SAS Institute. (n.d.-a). Feature index - desirability profiling (Optimization). Retrieved May 23, 2017, from https://www.jmp.com/en_us/software/feature-index.html#D

SAS Institute. (n.d.-b). Overview of stepwise regression. Retrieved May 23, 2017, from http://www.jmp.com/support/help/Overview_of_Stepwise_Regression.shtml

SAS Institute. (n.d.-c). Visual statistical data analysis tool (Version JMP Pro 13.1.0 (64 bit) [Computer software].

Shewry, M. C., & Wynn, H. P. (1987). Maximum entropy sampling. *Journal of Applied Statistics*, 14, 165–170.

Shockley, W., & Queisser, H. J. (1961). Detailed balance limit of efficiency of p-n junction solar cells. *Journal of Applied Physics*, 32(3), 510–519.

SILVACO Inc. (2004). *Atlas User's Manual*. Santa Clara, CA: Author.

SILVACO Inc. (n.d.). Atlas: 2D and 3D device simulator software (Version 4.2.5.R) [Computer software].

Solar cell structure. (n.d.). Retrieved May 23, 2017, from www.pveducation.org/pvcdrom/solar-cell-structure

Solar cell efficiency limit. (n.d.). Retrieved May 28, 2017, from http://solarcellcentral.com/limits_page.html

Steinberg, D. M., & Lin, D. K. J. (2006). A construction method for orthogonal Latin hypercube designs. *Biometrika*, 93(2), 279–288.

Tsutagawa, M. H. (2013). *Genetic algorithm metaheuristic methodology and Silvaco Atlas to produce maximum power and efficiency solar cell designs*. Doctoral dissertation, Naval Postgraduate School, Monterey, CA.

Utzler, J. (2006). *Genetic algorithm based optimization of advanced solar cell designs modeled in Silvaco Atlas*. Master's thesis, Naval Postgraduate School, Monterey, CA.

U.S. Department of Energy. (n.d.). *The history of solar power*. Retrieved May 28, 2017, from https://www1.eere.energy.gov/solar/pdfs/solar_timeline.pdf.

Vieira, Jr., H., Sanchez, S., Kienitz, K. H., & Belderrain, M. C. N. (2011). Generating and improving orthogonal designs by using mixed integer programming. *European Journal of Operational Research*, 215(3), 629–638.

Vieira, H. (2012). NOB_Mixed_512DP [Excel spreadsheet]. Retrieved April 15, 2017, from <http://my.nps.edu/web/seed/software-downloads>

Ye, K. Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association: Theory and Methods*, 93(444), 1430–1439.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California